Robustness in Multi-Objective Submodular Optimization: a Quantile Approach

Cedric Malherbe [1] **Kevin Scaman** [2 3]

Abstract

The optimization of multi-objective submodular systems appears in a wide variety of applications. However, there are currently very few techniques which are able to provide a robust allocation to such systems. In this work, we propose to design and analyse novel algorithms for the robust allocation of submodular systems through lens of quantile maximization. We start by observing that identifying an exact solution for this problem is computationally intractable. To tackle this issue, we propose a proxy for the quantile function using a softmax formulation, and show that this proxy is well suited to submodular optimization. Based on this relaxation, we propose a novel and simple algorithm called SOFTSAT. Theoretical properties are provided for this algorithm as well as novel approximation guarantees. Finally, we provide numerical experiments showing the efficiency of our algorithm with regards to state-of-the-art methods in a test bed of real-world applications, and show that SOFTSAT is particularly robust and well-suited to online scenarios.

1. Introduction

The maximization of submodular functions appears in many practical scenarios, including feature selection (Das & Kempe, 2018), network monitoring (de Badyn & Mesbahi, 2016), news article recommendation (El-Arini et al., 2009), sensor placement (Tzoumas et al., 2016), influence maximization (Kempe et al., 2003) and document summarization (Lin & Bilmes, 2011). In these applications, submodularity allows the use of fast optimization algorithms while retaining strong theoretical guarantees (Krause & Golovin, 2014).

[1]Huawei Noah's Ark Lab [2]DI ENS, École normale supérieure, CNRS, INRIA, PSL University [3]This work was done while the author was working at Huawei.. Correspondence to: Cedric Malherbe <cedric.malherbe@huawei.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

While it is widely known that greedy methods work well for a single objective (Nemhauser et al., 1978), the problem becomes much harder when there are multiple objectives to handle. In this paper, we focus on the case where the system we wish to optimize is described by multiple submodular functions $F_1, \ldots, F_d$ and we wish to provide a robust allocation among the objectives. In practice, standard approaches (Kempe et al., 2003; Krause et al., 2008) to solve this problem consists of either aggregating the system through the (possibly weighted) average-case $(1/d)\sum_i F_i(S)$ or the worst-case $\min_i F_i(S)$ of the objectives. However, only focusing on the average-case can lead to many objectives with poor values, while considering the worst-case may be too pessimistic to reflect the overall system and drastically reduces their use in practice (more specifically when there are outliers that naturally present low values). In this work, we propose to tackle these limitations by following the risk literature (Ben-Tal & Teboulle, 2007; Rostek, 2010) and propose to use the maximization of the quantiles of the objectives $Q_p(F_1(S), \ldots, F_d(S))$. Indeed, maximizing the p-th quantile of the objectives has already been shown to be a reliable measure for robustness in the risk literature and allows to discard the $p \times d$ outliers of the system with the worst objectives from the optimization while still robustly allocating the resources among the remaining objectives. Unfortunately, the major drawback of this approach is that the quantiles do not preserve submodularity and their direct maximization is thus not trivial. In this work, we show how to tackle this problem by introducing a novel tool: biased expectations that act as proxies for quantiles and allows to design provable algorithms. Up to our knowledge, our work provides the first algorithm that is designed to solve the robust submodular problem through quantile maximization. More precisely, our contribution can be summarized as follows: (1) a novel formulation of the robust multi-objective submodular maximization problem through the lens of quantile optimization; (2) an introduction to biased expectations as generic aggregate functions for robustness as well as its generic properties; (3) a novel algorithm called SOFTSAT which relies on biased expectations and comes, to the best of our knowledge, with the first approximation guarantees for quantile robustness; (4) an application of SOFTSAT to three motivating real-world optimization problems in natural language processing, vision and graph illustrating that it outperforms state-of-the-art methods.

The rest of the paper is organized as follows. We start with definitions and preliminaries in Section 2. In Section 3, we introduce the concept of biased expectations and present its applications in submodular optimization. In Section 4, we introduce the SOFTSAT algorithm and its associated approximation guarantees. Finally, the empirical performance of our algorithm is compared in Section 5. All proofs can be found in the appendix.

Related work. Perhaps, the closest work related to our analysis is the work of (Krause et al., 2008) where they focus on the problem of maximizing the worst-case objective $F_{wc}(S) = \min_{i=1...d} F_i(S)$ where F_i are submodular. They show that this problem is inapproximable unless P = NP and provide an algorithm that sequentially runs a series of greedy algorithms over the proxy function of the minimum $S \mapsto (1/d)\sum_{i=1}^{d} \min(F_i(S), c)$ for some constant c. Using the results of (Wolsey, 1982), they show that their algorithm returns a set with a better worst-case objective value than the optimal solution (i.e. an approximation error of 1) by violating the cardinality constraint. In the continuation of this work, (Anari et al., 2019) considered the same problem under matroid constraints and provide an extension of the greedy algorithm that returns a union of $O(\log(d/\varepsilon))$ sets that provide an approximation error of $(1-\varepsilon)$ by relaxing the algorithm's constraint set. Interestingly, they also use a soft-min operator similar to ours to extend their algorithm to the adversarial case. At the same time, (Chekuri et al., 2009) proposed a randomized polynomial time algorithm with $(1 - 1/e - \varepsilon)$ approximation guarantee for the problem of finding a set S such that $F_i(S) \geq V_i$ for some values V_i set as input, as an application of a new technique for rounding over a matroid polytope called swap rounding. Later, (Udwani, 2018) improved the runtime of the algorithm using multiplicative-weight-updates and (Iyer, 2019) analyzed the worst and best case objective problem under a broad range of combinatorial constraints such as cardinality, knapsack and matroid constraints. Finally, a second line of works related to our analysis considers the problem of maximizing the conditional value at risk $\text{CVaR}_p(S) = \mathbb{E}[F(S,X)|F(S,X) < Q_p(F(S,X))]$ of a system $F(S,X)$ under uncertainty where X is a random variable. In (Wilder, 2018), they give a $(1 - 1/e)$-approximation algorithm for maximizing the CVaR of a continuous submodular function relying on gradient descent. In (Ohsaka & Yoshida, 2017), they consider a specific influence maximization problem for cascade models and use the greedy algorithm to provide guarantees on the CVaR. In (Maehara, 2015), they consider the discrete CVaR maximization problem and provide hardness results. Finally, the approach we propose is also closely related to the works on generalization of submodular functions such as (Das & Kempe, 2011; Ghadiri et al., 2020; 2021).

2. Problem Statement and Preliminaries

In this section, we introduce the problem considered in the analysis as well as preliminary results.

Setup. Let $\mathcal{V} = \{1, ..., n\}$ be a set of $n \geq 1$ elements, called the ground set and let $F_1, \ldots, F_d$ be a collection of $d \geq 2$ non-negative non-decreasing submodular set functions defined over the power set of $\mathcal{V}$. In this paper, we focus on the problem of finding a subset $S_K^\star \subseteq \mathcal{V}$ of at most K elements which maximizes the p-th quantile of the values of the objective functions for some parameter $p \in (0, 1)$, i.e.,

$$S_K^\star \in \underset{S \subseteq \mathcal{V}: |S| \leq K}{\operatorname{argmax}} \; Q_p(F_1(S), \ldots, F_d(S)) \tag{1}$$

where $Q_p(x) := \min\{u \in \mathbb{R} : p \leq (1/d) \cdot \sum_{i=1}^{d} \mathbb{I}\{x_i \leq u\}\}$ for all $x \in \mathbb{R}^d$ denotes the p-th quantile. Informally speaking, the p-th quantile of the collection of functions $F_1(S), \ldots, F_d(S)$ is the cut-off point that divides the collection in two parts: $\lceil p \times d \rceil$ functions that have values below or equal to this threshold and $d - \lceil p \times d \rceil$ functions that have values above this threshold. Thus, maximizing the p-th quantile allows to find the allocation for which this cut-off point is maximal, providing the best value to at least $(1-p)$ percent of the collection of d functions. In this paper, our goal is to develop an algorithm that approximates such a solution by making a minimum number of evaluations of the collection of functions $F(S) = [F_1(S), \ldots, F_d(S)]$.

Motivating Examples. We present here three applications that require a robust allocation of resources in multi-dimensional systems. **(1) Resource Allocation:** the first example corresponds to the case where we have a collection of d users (or players) and we wish to share at most K resources from the ground set $\mathcal{V}$ among the users. Then, if the utility function $F_i(S)$ of each user $i \in \{1, \ldots, d\}$ is described by a submodular function when a set $S \subseteq \mathcal{V}$ is selected, this problem consists in solving an instance of Problem (1) if one wishes to provide a fair allocation of the resources among the users except the $p \times d$ worst users. An instance of this problem in the telecommunication domain is described in the next paragraph. **(2) Multi-objective Optimization:** the second example corresponds to the plain multi-objective optimization problem (Ostfeld et al., 2008). In this case, a system is naturally described by a collection of submodular functions $F_1, \ldots, F_d$ and one wishes to have at least $(1-p)d$ objectives with a maximum minimal value with a maximum budget of K actions taken from the ground set $\mathcal{V}$. The balanced news feed problem of Section 5 is a practical example of this problem. **(3) Robustness:** the last application covered by this framework is the case where

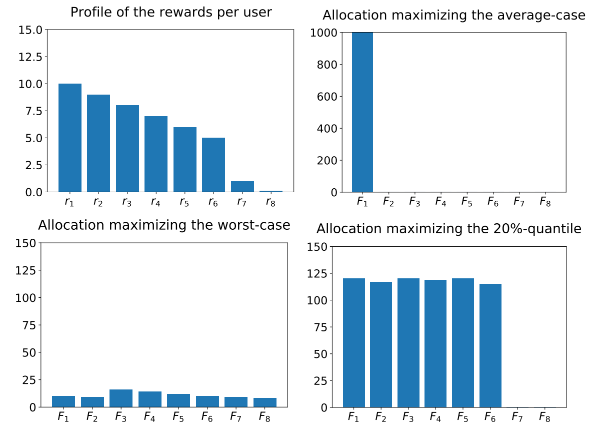

Figure 1. Profile of the average throughput per user and the allocations of resources maximizing the 20%-quantile, worst-case and average-case criterions. The first graph displays the rewards of the users. Note that the y-axis scales vary for each allocations.

there is a single submodular objective function $F(S, X)$ subject to uncertainty, captured by a random variable X. In this case, one often seeks to optimize over the worst-case realizations of the uncertain objective, resulting in an instance of multi-objective submodular maximization where the (potentially infinite) functions $F_i(S) = F(S, X_i)$ denote realizations of the uncertain objective. This covers the case of diffusion processes over graphs and Value-at-Risks (Kempe et al., 2003; Zhou & Tokekar, 2018).

Quantile vs. Worst and Average Cases. The key approach of the paper is to consider the quantile as a robust version of the worst-case criterion in multi-objective systems. This methodology of using quantiles instead of worst-cases and means has been widely used in mathematical finance and industrial science because they are less susceptible than means and worst-cases to explode or vanish under long-tailed distributions and outliers (Zhou & Tokekar, 2018; Wilder, 2018). Intuitively, quantiles provide a simple metric to find a good balance between the conflicting objectives of *average performance* and *robustness*. As an example, we consider the task of ensuring good communication quality over a wireless network. A natural objective is to ensure that all users enjoy the best communication quality possible. Unfortunately, users $i \in \{1, \dots d\}$ are not equal in their access to the wireless network due to their distance, and some may experience poor communication quality described by their average throughput r_i no matter how much effort is put into improving it. Each time the network allocates one resource to one user, it receives its average throughput r_i. Thus, the total signal received by user i when an allocation $S_K \subseteq \{1, \dots, n\}$ has been selected is defined by $F_i(S_K) = \sum_{e \in S_K} r_i \mathbb{I}\{e \mod d = i - 1\}$ which is submodular. Figure 1 displays the allocations among the different users obtained by maximizing the different criterions with $d = 8$ and $K = 100$. As it can be seen, the average-case criterion puts all its efforts on a single user and focusing on the worst-case would significantly downgrade the performance for all other users by giving the *entire* communication channel to this single user. Using quantiles allows to find a better balance between the overall performance of the network and equality of treatment between users, and is thus a common metric in real-world applications. Note that in real settings, the users with low values are paired with a different and closest station to ensure an overall good performance for all users.

Preliminaries. A natural approach to solve Problem (1) would be to perform an exhaustive search by evaluating the function $Q_p(F(S))$ over all the sets $S \subseteq \mathcal{V}$ of size K. However, this strategy would require to make $\binom{|\mathcal{V}|}{K}$ evaluations of the collection of functions $F(S) = [F_1(S), \dots, F_d(S)]$, which is often dissuasive in practice. Another more practical approach would be to use the greedy algorithm over the function $S \mapsto Q_p(F(S))$ to build an approximation of the solution $\hat{S}_K$ by making $\Omega(K \times |\mathcal{V}|)$ function evaluations. However, this strategy also presents some limitations. Indeed, following the arguments of (Krause et al., 2008), it can be shown that the quantile function does not preserve submodularity, and the classical $(1 - 1/e)$ approximation ratio cannot be achieved for $Q_p(F(\hat{S}_K))$ with greedy optimization, as shown below.

Proposition 2.1. *Consider any ground set $\mathcal{V}$ of $n \geq 4$ elements, any cardinality constraint $K \in \{2, \dots, |\mathcal{V}| - 2\}$ and any quantile parameter $p \in (0, 1 - 1/d)$. Then, for any constant $\varepsilon > 0$ arbitrarily small, there exists a collection $F_1, \dots, F_d$ of non-negative, monotone, submodular functions such that the greedy algorithm returns a set $\hat{S}_K$ satisfying:*

$$Q_p(F(\hat{S}_K)) \leq \varepsilon \times \max_{|S| \leq K} Q_p(F(S)) \,.$$

More strikingly, it can also be shown that it is not possible to design a universal algorithm that achieves a positive approximation ratio in polynomial time for Problem (1), unless P = NP.

Proposition 2.2. *Under the assumptions of Proposition 2.1, if there exists a positive function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ and an algorithm that is guaranteed to find a set $\hat{S}_K$ of size K such that for all n, K and d, in time polynomial to the problem instance n we have $Q_p(F(\hat{S}_K)) \geq \gamma(n) \cdot \max_{|S| \leq K} Q_p(F(S))$, then P = NP.*

To tackle this limitation, we propose in this work to relax the optimization objective by using biased expectations as a proxy for the quantiles.

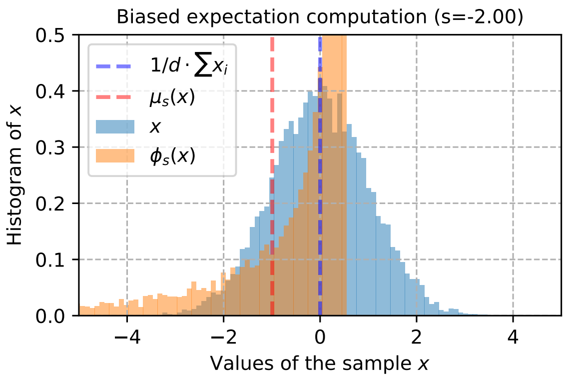

Figure 2. Computation of the biased expectation over a sample $x = (x_1, \dots, x_d)$ of d independant Gaussian random variables with $s = -2$.

3. Approximation of the Quantiles

As shown in Section 2, the quantiles do not preserve submodularity and are ill-suited to greedy algorithms. We thus propose to approximate quantiles via *biased expectations*, a notion first introduced in the context of non-convex optimization by (Scaman & Malherbe, 2020), and connected to softmax operators (Boyd & Vandenberghe, 2004).

Definition 3.1 (Biased Expectation)**.** Let $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ be a sample of $d \in \mathbb{N}^\star$ real values. Then, for any $s \in \mathbb{R}$, we define the biased expectation of the sample x with parameter s as follows:

$$\mu_s(x) = \phi_s^{-1}\left(\frac{1}{d}\sum_{i=1}^{d}\phi_s(x_i)\right)$$

where $\phi_s(x) = (e^{sx} - 1)/s$ if $s \neq 0$ and x otherwise.

This criterion is a particular instance of *quasi-arithmetic means* first analyzed by (Hardy et al., 1952), and is tightly connected to the cumulant generating function $\kappa_x(s) = \log((1/d)\sum_{i=1}^d e^{sx_i})$ and the LogSumExp operator $LSE(x) = \log(\sum_{i=1}^d e^{x_i})$. It can be seen as a generalization of the standard expectation $(1/d) \cdot \sum_{i=1}^d x_i$ parametrized by a function $\phi_s(\cdot)$. Indeed, $\mu_0(x) = (1/d) \cdot \sum_{i=1}^d x_i$, and ϕ_s will put more weight on the large values of the sample when $s > 0$ and less weight on large values when $s < 0$, hence the name *biased* expectation. A visual example of the computation of biased expectations is provided in Figure 2.

3.1. Generic Properties of Biased Expectations

Biased expectations play a central role in our analysis as they play the role of a smooth proxy function for the quantile function $Q_p(\cdot)$. These connections can be seen in the following property:

Proposition 3.2 (Biased expectation specific values)**.** *Let $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ be any sample of real values. Then, the function $s \mapsto \mu_s(x)$ is continuous and non-decreasing. Moreover, $\lim_{s\to-\infty}\mu_s(x) = \min_{i=1\dots n} x_i$, $\lim_{s\to 0}\mu_s(x) = \frac{1}{d}\sum_{i=1}^d x_i$ and $\lim_{s\to+\infty}\mu_s(x) = \max_{i=1\dots d} x_i$.*

Thus, biased expectations interpolate between the minimum (when $s \to -\infty$) and maximum (when $s \to +\infty$) of the sample $(x_1, \dots, x_d)$ depending on the value of s. A crucial consequence of the latter observation is that for any quantile parameter $p \in (1/d, 1 - 1/d)$, one can recover the quantile of the sample x by observing that there exists $s^\star \in \mathbb{R}$ such that $Q_p(x) = \mu_{s^\star}(x)$. Keeping this observation in mind, we also cast an intermediate result about quantile control.

Proposition 3.3 (Control of quantiles)**.** *Let $x = (x_1, \dots, x_d) \in \mathbb{R}_+^d$ be a sample of non-negative values. Then, for any $s \in \mathbb{R}^\star$ and $p \in (0, 1)$, we have:*

$$Q_p(x) \leq \frac{1}{s}\log\left(1 + \frac{e^{s\mu_s(x)} - 1}{1 - p}\right).$$

3.2. Properties in Submodular Optimization

Now, we present the advantages of using biased expectations in the case of submodular optimization. First, we show that, similarly to quantiles, using biased expectation allows to have a solution that interpolates between the solution of the worst-case and average-case problem.

Proposition 3.4. *Let $F_1, \dots, F_d$ be a collection of non-negative monotone submodular functions and define $f : s \mapsto \max_{|S|\leq K}\mu_s(F(S))$. Then, f is non-decreasing, $\lim_{s\to-\infty} f(s) = \max_{|S|\leq K}(\min_{i=1\dots d} F_i(S))$ and $\lim_{s\to 0} f(s) = \max_{|S|\leq K}(\frac{1}{d}\sum_{i=1}^d F_i(S))$.*

Thus, it is possible to find a balance between the worst-case and average-case solutions by tuning the parameter s and maximizing the biased expectation of the objectives. Finally, the last key characteristic of biased expectations that makes them suitable for submodular optimization is that—as opposed to quantiles—they preserve submodularity in the following sense:

Proposition 3.5 (Submodularity conservation)**.** *Let $F_1, \dots, F_d$ be non-decreasing, non-negative submodular functions and $s < 0$ negative. Then, the function $\mu_s(F(S)) = g(h(S))$ where g is non-negative and increasing, and h is a non-decreasing non-negative submodular function.*

This result shows that the problem of finding a set maximizing biased expectations is equivalent to the problem of finding a set maximizing a single submodular function h (since $\arg\max_{|S|\leq K} g(h(S)) = \arg\max_{|S|\leq K} h(S)$), and can thus be done using the greedy algorithm.

Algorithm 1 SOFTSAT

Require: Ground set $\mathcal{V}$, cardinality constraint K, functions $F(\cdot) = [F_1(\cdot), \ldots, F_d(\cdot)]$, parameter $s \in \mathbb{R}$

```
1: Ŝ_0 ← ∅
2: for t = 0 to K − 1 do
3:     e_{t+1} ← arg max_{e∈V/Ŝ_t} μ_s(F(Ŝ_t ∪ {e}))
4:     Ŝ_{t+1} ← Ŝ_t ∪ {e_{t+1}}
5: end for
6: return Ŝ_K
```

Remark 3.6 (Positive Values of s). We point out that the results presented in the paper can be adapted to the case $s > 0$. In this case, it can be shown that the function $h(S)$ exhibited in Proposition 3.5 is not submodular, but has a submodularity ratio $\gamma \geq \min_{i=1\ldots d}(s \sum_{e\in\mathcal{V}} F_i(e))/(e^{s\sum_{e\in\mathcal{V}} F_i(e)} - 1)$ which enjoys an approximation error of $(1 - e^{-\gamma})$ for greedy optimization (Das & Kempe, 2018). Thus, the results presented below can be extended to the case where $s > 0$ at a performance cost, but we only focus here on the results for $s < 0$ for simplicity and since in most application we want an allocation more robust than the allocation maximizing the average case ($s = 0$).

4. The SOFTSAT Algorithm

In this section, we introduce the SOFTSAT algorithm and provide its approximation guarantees.

4.1. Definition of SOFTSAT

The SOFTSAT algorithm (Algorithm 1) takes as input the ground set $\mathcal{V}$, the cardinality constraint K, the collection of functions $F(S) = [F_1(S), \ldots, F_d(S)]$ and a real-valued parameter s. It returns a set $\hat{S}_K \subseteq \mathcal{V}$ of cardinality at most K as an output. The algorithm proceeds sequentially in order to build the output set. It starts with an empty set $\hat{S}_0 = \emptyset$ and adds, at each iteration $t \geq 0$, a single element e_{t+1} among the set of elements that have not been selected yet $\mathcal{V}/\hat{S}_t$ (cf. line 3 and 4). To explain the selection process, observe first that since the arg-max function is stable by constant addition, then we have $e_{t+1} \in \arg\max_{e\in\mathcal{V}/\hat{S}_t} \mu_s(F(\hat{S}_t \cup \{e\})) = \arg\max_{e\in\mathcal{V}/\hat{S}_t} \mu_s(F(\hat{S}_t \cup \{e\})) - \mu_s(F(\hat{S}_t))$. Thus, at each iteration, the algorithm adds the element e_{t+1} from the remaining set which provides the best improvement over the current marginal value of the biased expectation $\Delta_{\mu\circ F}(e|\hat{S}_t) = \mu_s(F(\hat{S}_t \cup e)) - \mu_s(F(\hat{S}_t))$, which corresponds to the loop of the greedy algorithm (Nemhauser et al., 1978). In other words, the main idea behind the SOFTSAT algorithm is (1) to turn a multi-dimensional problem defined by the components $F_1, \ldots, F_d$ into a single objective optimization problem using biased expectations as a an aggregate function and (2) using the greedy algorithm to optimize this objective by sequentially maximizing the marginal value of the biased expectation.

Remark 4.1 (Connection with SATURATE). By Proposition 3.5, we know that $\mu_s(F(S)) = g(\sum_{i=1}^d \phi_s(F_i(S)))$ where g is non-decreasing. Hence, running SOFTSAT is equivalent to running the greedy algorithm over the function $\sum_{i=1}^d \phi_s(F_i(S))$. Now, observing that (1) the function $x \mapsto \phi_s(x)$ can be seen as a smooth approximation of the function $x \mapsto \min(x, -1/s)$ and (2) SATURATE (Krause et al., 2008) runs a series of greedy algorithms over the aggregate function $\sum_{i=1}^d \min(F_i(S), 1/s)$ for some $s > 0$, it is easy to see that SOFTSAT can be seen as a soft version of SATURATE.

Remark 4.2 (Computational Aspects). Running SOFTSAT with parameter K requires to make $\Omega(K \times |\mathcal{V}|)$ calls to the oracle $F(\cdot)$, which can be prohibitive when the cardinality of the ground space $|\mathcal{V}|$ becomes too large. Up to our knowledge, two solutions can be used to reduce its computational time in practice. The first solution consists of maintaining an upper bound of the marginal gains for each element $e \in \mathcal{V}$ by using the fact that $\Delta_F(e|S_{t+1}) \leq \Delta_F(e|S_t)$ by submodularity (see, e.g., (Minoux, 1978)), allowing to make fewer calls to the oracle F. Second, another solution consists of randomly sampling only $|\mathcal{V}|/K \log(1/\varepsilon)$ elements from the remaining set in the maximization process (line 3) for a given $\varepsilon > 0$ and selecting the one that provides the best marginal improvement, allowing to reduce the complexity to $\Omega(|\mathcal{V}| \times \log(1/\varepsilon))$ at a performance cost (see, e.g., (Mirzasoleiman et al., 2015)).

4.2. Approximation Guarantees

Recall first that the problem of finding a set that presents a constant approximation error with regards to the optimal quantile is not feasible unless P = NP (Proposition 2.2). We tackle this issue by: (1) first analyzing biased expectations instead of quantiles, (2) adding assumptions on the optimal parameter s and (3) finally relaxing the cardinality constraint K to obtain universal bounds.

Control on Biased Expectations. We start to provide an approximation guarantee between the difference of the result provided by SOFTSAT and the value of the optimal biased expectation.

Proposition 4.3 (Biased Expectation Approximation)**.** *Let $F_1, \ldots, F_d$ be a collection of d non-negative monotone submodular functions and fix any $K \geq 1$. Then, if $\hat{S}_K$ denotes the output of the SOFTSAT algorithm tuned with parameter $s < 0$, we have the following guarantee:*

$$\mu_s(F(\hat{S}_K)) \geq \left(1 - \frac{1}{f(s)}\right) \cdot \max_{|S|\leq K} \mu_s(F(S))$$

where the function $f(s) = -s\bar{\mu}/\log(1/e e^{-s\bar{\mu}} + (1 - 1/e))$ is strictly increasing and we have $\lim_{s\to 0} f(s) = e$ and $\lim_{s\to -\infty} f(s) = 1$ where $\bar{\mu} = \max_{|S|\leq K} \sum_{i=1}^d F_i(S)/d$.

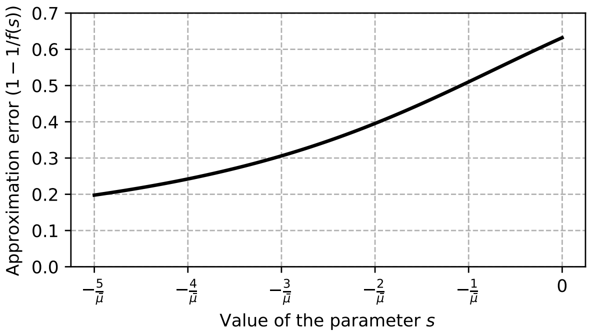

Figure 3. Values of the approximation error $(1-1/f(s))$ provided in the bound of Proposition 4.3 for different values of s.

This result shows that SOFTSAT does indeed provide an approximation of the maximizer of the biased expectation tuned with the same parameter s. The control provided is similar (in form) to the one obtained with a greedy algorithm over a single submodular function where the approximation error is equal to $(1-1/e)$ (see (Nemhauser et al., 1978)). However, since here the function $\mu_s(\cdot)$ is a composition of a strictly increasing function and a submodular function (Proposition 3.5), the approximation error varies with the value of the parameter s and the quantity $\bar{\mu}$ which is problem-specific. As an example, the values of the approximation error $(1-1/f(s))$ are displayed in Figure 3. It is interesting to note that when $s \to 0$, we recover the standard approximation error $(1-1/e) \approx 0.63$, which can be explained by the fact that $\mu_s(F(S)) \to (1/d)\sum_{i=1}^{d} F_i(S)$. Thus, using biased expectation allows to have a non-zero approximation ratio below the average-case regime, as opposed to the direct use of quantiles.

Control on Quantiles (Additional Assumptions). In the continuation of Proposition 4.3, we now show that one can recover a bound for the quantiles given a proper choice of parameter s. Indeed, as discussed in Section 3, biased expectations can approximate quantiles, and, for any $p \in (0,1)$ and $x \in \mathbb{R}^d$, there exists $s^* \in \mathbb{R} \cup \{-\infty, +\infty\}$ such that $\mu_s^*(x) = Q_p(x)$, which allows to obtain:

Corollary 4.4. *Let $S_K^\star \in \arg\max_{|S|\leq K} Q_p(F(S))$ be a solution of Problem (1) with $p \in (1/d, 1-1/d)$ and let $s^\star$ be a value such that $\mu_{s^\star}(F(S_K^\star)) = Q_p(F(S_K^\star))$. Then, assuming that $s^\star < 0$, the SOFTSAT algorithm tuned with parameter $s^\star$ returns a set $\hat{S}_K$ satisfying:*

$$Q_p(F(\hat{S}_K)) \geq \hat{C} \cdot \left(1 - \frac{1}{f(s^\star)}\right) \cdot \max_{|S|\leq K} Q_p(F(S))\,,$$

where $\hat{C} = \min_i F_i(\hat{S}_K)/\max_i F_i(\hat{S}_K)$, and $f(\cdot)$ is defined in Proposition 4.3.

This result shows that SOFTSAT does provide an approximation of the maximizer of the quantile, given a proper choice for the parameter s. In addition to Proposition 4.3, this result involves problem-specific constants through the terms $s^\star$, $\bar{\mu}$ and also $\hat{C}$. Here, the constant $\hat{C}$ controls the spread of the function values of $F(\hat{S}_K)$ after a run, and will be large for particularly symmetric problems. Thus, this results allows to recover approximation results for an inapproximable problem (i.e. Proposition 2.2) at the extra cost that (1) it uses a parameter $s^\star$ that requires the knowledge of the optimal set $S_K^\star$ and (2) its approximation ratio depends on the output set via the term $\hat{C}$. We will see in the next section how to select a parameter s in practice and below how to obtain a universal constant that is not problem-specific.

Control on Quantiles (Cardinality Relaxation). Finally, similary to (Krause et al., 2008), we show that relaxing the cardinality constraint (which is generally not possible in practice) allows to achieve a universal approximation ratio for the quantile problem. The next result shows that an approximation ratio of order $\Omega(1/d\ln(d)p)$ can be achieved if we allow the set to be logarithmically larger than the optimal solution.

Proposition 4.5. *Let $S_K^\star \in \arg\max_{|S|\leq K} Q_p(F(S))$ be a solution of Problem (1) with $p \in [1/d, 1-1/d]$ and set $s^\star = -\ln(4d)/Q_p(F(S_K^\star))$. Then, SOFTSAT tuned with cardinality constraint $K' = K\lceil\ln(4d)\rceil$ and $s^\star$ finds a solution $\hat{S}_{K'}$ such that*

$$Q_p(F(\hat{S}_{K'})) \geq \frac{\ln(2)}{\lceil pd\rceil \ln(4d)} \cdot \max_{|S|\leq K} Q_p(F(S))\,.$$

Interestingly, this result suggests that independently of the functions $F_1, \ldots, F_d$ the quantile maximization problem becomes linearly harder as the quantile parameter p is small and linearly harder with the dimensionality d. It is also interesting to compare this result to that of (Krause et al., 2008) which obtains an approximation ratio of 1 for the worst-case problem ($p = 1/d$) using approximately $\ln(d)$ times as many elements as in the optimal solution at making $O(n^2 d\log(d))$ function evaluations. Here, for the worst-case, we obtain an approximation ratio of $\Omega(1/\ln(4d))$ using approximately $\ln(d)$ times as many elements and making $O(n^2)$ function evaluations. Thus, for the worst-case, we have a trade-off and using biased expectations instead of SATURATE allows to make less function evaluations at the cost of a lower approximation error. Lastly, recall also that SOFTSAT is designed to optimize the whole quantiles as opposed to SATURATE which only focuses on worst-cases.

4.3. Choice of the Parameter s in Practice

Here, we present two heuristics that can be used to tune the parameter s in practice.

Quantile as Threshold. The first method consists of using the estimate $s = -1/Q_p(F(S_K^\star))$ which comes from Proposition 4.5 and the fact that the function $x \mapsto 1 - e^{sx}$ used in biased expectations can be seen as a soft approximation of the function $x \mapsto \min(-sx, 1)$ that plays the role of threshold function at $-1/s$. Thus, setting the threshold around the quantile $Q_p(F(S^\star_K))$ helps to prevent the algorithm to put too much effort a single objective that already reached the specified quantile value. This estimate can be computed from (1) the result of a previous experiments that provided a set $\hat{S}$ and then setting $s = -1/Q_p(F(\hat{S}))$ or (2) in a online fashion by setting at each iteration $s_t = -1/Q_p(F(\hat{S}_t))$ where $\hat{S}_t$ denotes the set computed at each iteration in Algorithm 1.

Grid Search. The second method we propose consists of evaluating the performance of the algorithm for several values for s that belong to a grid such as $G = [-100, -10, -1, -0.1, -0.01]$ and keeping the value of s that provides the best results.

In the experiments, we considered both (1) an adaptive version of SOFTSAT which estimates s along with the optimization and has complexity $\Omega(K \times |\mathcal{V}|)$ and (2) a version that estimates s with a grid search and has complexity $\Omega(|G| \times K \times |\mathcal{V}|)$ where $|G|$ is the size of the grid.

5. Numerical Eexperiments

In this section, we compare the empirical performance of the SOFTSAT algorithm.

Algorithms. Seven different algorithms that are commonly used to solve multi-objective submodular problems were used in the benchmark. **SOFTSAT** is displayed in Algorithm 1 and we performed a grid search for the parameter s over the grid $G = [-100, -10, -1, -0.1, -0.01]$. **AdaSOFTSAT** is the adaptive version of SOFTSAT, where we start with the value $s_0 = 0$ and then update at each iteration the value of the parameter s_t using the rule: $s_t = -1/Q_p(F(\hat{S}_t))$ described in Section 4.3. **SATURATE** (Krause et al., 2008) which performs a bisection over the value of $\max_{|S| \leq K} \min_{i=1 \dots d} F_i(S)$ by running a series of greedy algorithms. This algorithm is specifically designed to solve the $1/d$-th quantile problem. **AdaSATURATE** which is an adaptive version of SATURATE and uses at each step $Q_p(\hat{S}_t)$ as an estimate of the threshold. It is used to compare the performance of AdaSOFTSAT. **GreedyQuantile** which consists of running the standard greedy algorithm (Nemhauser et al., 1978) over the quantile $Q_p(F(S))$. **GreedyMean** which consists of running the greedy algorithm over the average-case $(1/d) \cdot \sum_{i=1}^d F_i(S)$. **Random** which performs a series of random set evaluations. An implementation of the algorithms can be found in the Appendix.

Balanced News Feed. The first problem consists in selecting a subset of news articles that fairly covers a given number of topics. More formally, we have access to a set $\mathcal{V} = \{1, \dots, n\}$ of n articles that cover d different topics (sport, politics, economy, etc.). For each article $e \in \mathcal{V}$, we have a feature vector $f(e) = [f_1(e), \dots, f_d(e)] \in [0, 1]^d$ that represents the proportion of the article that is dedicated to each topic $i \in \{1, \dots, d\}$. For instance, we may have $f(1) = [0.1, 0.1, 0, 0, 0, 0.9]$ and $f(2) = [0.3, 0.1, 0.3, 0, 0, 0.9]$ where the summation can be ≥ 1 due to overlapping topics. Given a subset of articles $S \subseteq \mathcal{V}$, the proportion of the articles that cover a given topic i is given by

$$F_i(S) = \sum_{e \in S} f_i(e)$$

which is a non-negative monotone (sub)modular function. Hence, if one wants to select a subset of K articles that cover fairly at most $p \times d$ topics, one can solve the problem $\max_{|S| \leq K} Q_p(F_1(S), \dots, F_d(S))$ which is an instance of Problem (1). In practice, we took the BBC news data set (Greene & Cunningham, 2006) that contains $n = 2250$ articles and we used the uClassify topic classification API (Kagström et al., 2014) to obtain the feature vectors with $d = 10$ topics for each article.

Images Summarization. The second task consists in selecting a subset of K images from a collection $\mathcal{V} = \{1, \dots, n\}$ of n images that fairly represents most of the images without considering outliers (images that appear few times or are not relevant in the dataset). More formally, given a subset of images $S \subseteq \mathcal{V}$ and a non-negative distance $D(\cdot, \cdot)$ over images, one can measure if an image $i \in \mathcal{V}$ is well taken into account in a subset $S \subseteq \mathcal{V}$ by measuring the similarity to the closest image to the subset S through the function

$$F_i(S) = 1 - \min_{e \in S} D(i, e)$$

which is non-negative, monotone and submodular. Hence, if one wants to have a subset of K images that represents most the dataset except the $p \times n$ outliers, one can solve the problem $\max_{|S| \leq K} Q_p(F_1(S), \dots, F_d(S))$ where p correspond to the proportion of the images we want to discard. In practice, we used the pokemon dataset (Churchill, 2017) that contains $n = d = 151$ images and we used the normalized cosine similarity $D(i, e) = v_i \cdot v_e / |v_i \cdot v_e| + \min_{e, e' \in \mathcal{V}^2} v_e \cdot v_{e'} / |v_e \cdot v_{e'}|$ where the vectors v_e are the centered and normalized img2vec embeddings computed from an AlexNet tuned over the ImageNet data set (Krizhevsky et al., 2017) and the second term is added to prevent non-negativity.

Graph Covering. The last application consists in selecting a subset of source nodes from a graph that can fairly communicate with the best $(1 - p)$ percentage of the remaining nodes of the graph. More formally, let $G = (\mathcal{V}, \mathcal{E})$ be a connected graph, let $D(\cdot, \cdot)$ denotes the minimal distance

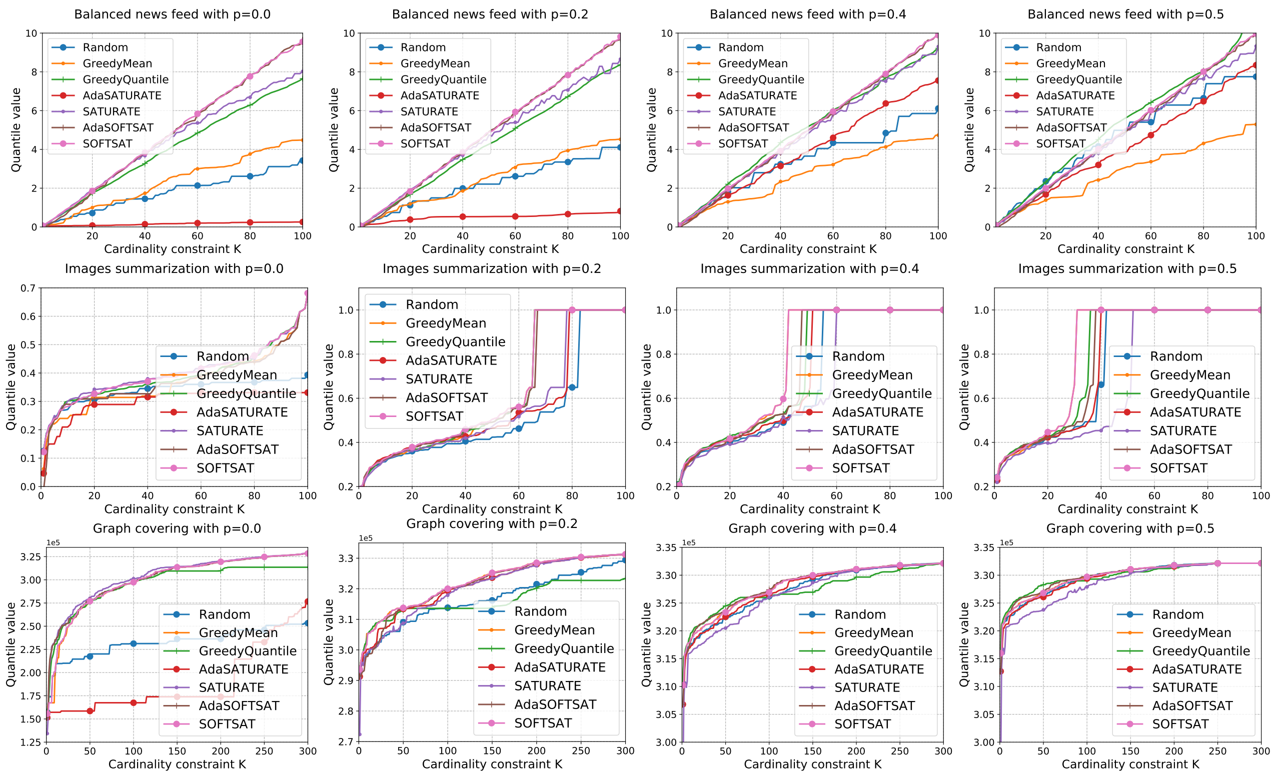

Figure 4. Results of the numerical experiments. The graph displays the values of quantiles $Q_p(F(\hat{S}))$ provided by each algorithm over the test problems of the benchmark for different values of K and p.

between two nodes. Then, if the ground set is the set of all nodes $\mathcal{V} = \{1, \ldots, n\}$ and one selects a subset $S \subseteq \mathcal{V}$ of source nodes from the graph, the minimum distance from the subset of source nodes S to a given node $i \in \mathcal{V}$ is given by $S \mapsto \min_{e \in S} D(e, i)$ which is non-increasing and supermodular (i.e. $-F$ is submodular). To ensure non-negativity and submodularity, one can thus use the function

$$F_i(S) = \max_{(e,e') \in \mathcal{V}^2} D(e, e') - \min_{e \in S} D(e, i).$$

Hence, if one wants to select a subset of K source nodes that can fairly communicate with $1 - p$ percent nodes of the graph, one can solve the problem $\max_{|S| \le K} Q_p(F_1(S), \ldots, F_d(S))$ where p correspond to the proportion of the nodes we want to discard. In practice, we used the USAirport 500 dataset (Colizza et al., 2007) which is a graph that contains the 500 largest commercial airports in the United States.

Protocol. For each problem, we ran the seven algorithms with different values of $p \in \{0, 0.2, 0.4, 0.5\}$ and $K \in \{1, \ldots, K_{\max}\}$ where $K_{\max} = 100$ (balanced news feed), 100 (images summarization) and 300 (graph covering). For each algorithm A and pair of values p and K, we recorded the value of the quantile $Q_p(F(\hat{S}_{A,p,K}))$ provided by each algorithm where $\hat{S}_{A,p,K}$ denotes the output of the algorithm A with values set to K and p. Results are collected in Figure 4.

Remarks. First, observe that, independently of the algorithm, considering quantile maximization (i.e. $p > 0$) instead of the worst-case maximization ($p = 0.0$) can drastically improve the optimization performance. For instance, in images summarization, the best algorithm can only reach the value of 0.7 with $K = 100$ images when $p = 0.0$, while discarding 20% of outliers ($p = 0.2$) allows to reach the maximum value of 1.0 with only $K = 64$ images. Second, it is interesting to note that as the value of p increases, the random strategy tends to match the best algorithms (e.g., graph covering and balanced news feed when $p = 0.5$). This can be interpreted as the fact that the problem becomes harder as the quantiles grow large. Last, as discussed in Section 2, observe that directly maximizing the quantiles of the objectives—that do not preserve submodularity— with the greedy algorithm (GreedyQuantile) does not generally lead to optimal performance (e.g., graph covering with $p = 0.2$ where quantile maximization performs worst than the random strategy). As one can see, the SOFTSAT algorithm displays consistent results over the three test problems of the benchmark in the sense that it constantly achieves the best performance (e.g., balanced news feed, images summarization and graph covering with $p = 0.2$), while most methods fail on at least one problem (e.g., GreedyMean on balanced news feed and GreedyQuantile on graph covering). More strikingly, for the images summarization problem with $p = 0.4$ and $p = 0.5$, the SOFTSAT algorithm only requires $K = 41$ and $K = 36$ images to successfully summarize the dataset (i.e. reaching 1) while other methods may require up to $K = 50$ images. Moreover, it is also interesting to note that the adaptive version of the algorithm with a lower complexity of $K \times |V|$ performs similarly to SOFTSAT in most test problems (e.g., balanced news feed and graph covering). More specifically, for the worst-case (i.e. $p = 0$), SOFTSAT algorithm displays similar results to SATURATE which is specifically tailored to solve this problem, while AdaSOFTSAT always outperfoms AdaSATURATE, validating the fact that using soft approximations lead to more robust results in online settings.

6. Conclusion

In this paper, we introduced a robust formulation of the multi-objective submodular maximization problem through the use of quantiles as well as the concept of biased expectations. We showed how to adapt the greedy algorithm with biased expectation, leading us to an efficient algorithm assessed in a test bed of real-world applications.

References

Anari, N., Haghtalab, N., Naor, S., Pokutta, S., Singh, M., and Torrico, A. Structured robust submodular maximization: Offline and online algorithms. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3128–3137. PMLR, 2019.

Ben-Tal, A. and Teboulle, M. An old-new concept of convex risk measures: The optimized certainty equivalent. *Mathematical Finance*, 17(3):449–476, 2007.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, March 2004. ISBN 0521833787.

Chekuri, C., Vondrák, J., and Zenklusen, R. Dependent randomized rounding for matroid polytopes and applications. *arXiv preprint arXiv:0909.4348*, 2009.

Churchill, S. Pokemon images dataset, 2017. URL https://www.kaggle.com/kvpratama/pokemon-images-dataset.

Colizza, V., Pastor-Satorras, R., and Vespignani, A. Reaction–diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics*, 3(4): 276–282, 2007.

Das, A. and Kempe, D. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

Das, A. and Kempe, D. Approximate submodularity and its applications: subset selection, sparse approximation and dictionary selection. *The Journal of Machine Learning Research*, 19(1):74–107, 2018.

de Badyn, M. H. and Mesbahi, M. Growing controllable networks via whiskering and submodular optimization. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 867–872. IEEE, 2016.

El-Arini, K., Veda, G., Shahaf, D., and Guestrin, C. Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 289–298, 2009.

Ghadiri, M., Santiago, R., and Shepherd, B. A parameterized family of meta-submodular functions. *arXiv preprint arXiv:2006.13754*, 2020.

Ghadiri, M., Santiago, R., and Shepherd, B. Beyond submodular maximization via one-sided smoothness. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1006–1025. SIAM, 2021.

Greene, D. and Cunningham, P. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pp. 377–384, 2006.

Hardy, G., Collection, K. M. R., Littlewood, J., Pólya, G., Pólya, G., and Littlewood, D. *Inequalities*. Cambridge Mathematical Library. Cambridge University Press, 1952. ISBN 9780521358804.

Iyer, R. A unified framework of robust submodular optimization. *arXiv preprint arXiv:1906.06393*, 2019.

Kagström, J., Karlsson, R., and Kagström, E. Uclassify, 2014.

Kempe, D., Kleinberg, J., and Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, 2003.

Krause, A. and Golovin, D. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pp. 71–104. Cambridge University Press, 2014.

Krause, A., McMahan, H. B., Guestrin, C., and Gupta, A. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec):2761–2801, 2008.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 510–520, 2011.

Maehara, T. Risk averse submodular utility maximization. *Operations Research Letters*, 43(5):526–529, 2015.

Minoux, M. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization techniques*, pp. 234–243. Springer, 1978.

Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., and Krause, A. Lazier than lazy greedy. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 1812–1818, 2015.

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.

Ohsaka, N. and Yoshida, Y. Portfolio optimization for influence spread. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 977–985, 2017.

Ostfeld, A., Uber, J. G., Salomons, E., Berry, J. W., Hart, W. E., Phillips, C. A., Watson, J.-P., Dorini, G., Jonkergouw, P., Kapelan, Z., et al. The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms. *Journal of Water Resources Planning and Management*, 134(6):556–568, 2008.

Rostek, M. Quantile maximization in decision theory. *The Review of Economic Studies*, 77(1):339–371, 2010.

Scaman, K. and Malherbe, C. Robustness analysis of non-convex stochastic gradient descent using biased expectations. *Advances in Neural Information Processing Systems*, 33, 2020.

Tzoumas, V., Jadbabaie, A., and Pappas, G. J. Sensor placement for optimal kalman filtering: Fundamental limits, submodularity, and algorithms. In *2016 American Control Conference (ACC)*, pp. 191–196. IEEE, 2016.

Udwani, R. Multi-objective maximization of monotone submodular functions with cardinality constraint. *Advances in Neural Information Processing Systems*, 31: 9493–9504, 2018.

Wilder, B. Risk-sensitive submodular optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Wolsey, L. A. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4): 385–393, 1982.

Zhou, L. and Tokekar, P. An approximation algorithm for risk-averse submodular optimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pp. 144–159. Springer, 2018.

A. Algorithms Used in the Benchmark

Here, we provide the detailed implementations of the algorithms considered in the benchmark. We start by giving the definition of the simple greedy algorithm (Algorithm 2).

Algorithm 2 Greedy algorithm

Require: Ground set $\mathcal{V}$, cardinality constraint K, a set function $F : 2^{\mathcal{V}} \to \mathbb{R}$

$\hat{S}_0 \leftarrow \emptyset$
for $t = 0$ to $K-1$ **do**
  $e_{t+1} \leftarrow \arg\max_{e \in \mathcal{V}/\hat{S}_t} F(\hat{S}_t \cup \{e\}) - F(\hat{S}_t)$
  $\hat{S}_{t+1} \leftarrow \hat{S}_t \cup \{e_{t+1}\}$
end for
return $\hat{S}_K$

In the benchmark, the GreedyMean algorithm consists in running Algorithm 2 over the average-value of the objectives $F_{ac} : S \mapsto (1/d) \cdot \sum_{i=1}^{d} F_i(S)$, while the GreedyQuantile consists in running Algorithm 2 over the quantile of the objectives $S \mapsto Q_p(F_1(S), \dots, F_d(S))$. Note that the GreedyMean algorithm does not depend on the value of p. On the other hand, the RandomQuantile algorithm is defined as follows (see Algorithm 3). Remark that it makes exactly the same number of calls to the function $F(S)$ than the standard greedy algorithm which is $\Omega(K \times |\mathcal{V}|)$.

Algorithm 3 RandomQuantile

Require: Ground set $\mathcal{V}$, cardinality constraint K, submodular functions $F(\cdot) = [F_1(\cdot), \dots, F_d(\cdot)]$, parameter $p \in [0, 1]$

$\hat{S}_K \leftarrow \emptyset$
$Q_{\max} \leftarrow 0$
for $t = 0$ to $K-1$ **do**
  for $i = 1$ to $|\mathcal{V}| - t$ **do**
    $S_{test} \leftarrow \mathcal{U}(\{S \subseteq \mathcal{V} : |S| = i\})$
    if $Q_p(F(S_{test})) \geq Q_{\max}$ **then**
      $\hat{S}_K \leftarrow S_{test}$
      $Q_{\max} \leftarrow Q_p(F(S_{test}))$
    end if
  end for
end for
return $\hat{S}_K$

Finally, for the SATURATE algorithm, we used the same implementation as the one provided in (Krause et al., 2008) with the parameter α set to 1 as advised in the paper. Note that for this algorithm the author provide an upper bound on its complexity that is of order $O(|\mathcal{V}|^2 \times d \times \log(d \min_i F_i(\mathcal{V})))$. The implementation we used can be found in Algorithm 4.

B. Proofs of the Theoretical Results

In this section, we provide the detailed proofs of the results presented in the paper.

Proof of Proposition 2.1. Consider without loss of generality the ground set $\mathcal{V} = \{e_1, \dots, e_{|\mathcal{V}|}\}$, set any constant $C > 1$, set $i^\star = \lceil (1-p)d \rceil$ and let $F_1, \dots, F_d$ be the collection of $d \geq 2$ non-negative, monotone (sub)modular functions defined as follows:

$$F_i(S) = \sum_{e \in S} F_i(\{e\})$$

where for the first element e_1, we have:

$$F_i(\{e_1\}) = \begin{cases} C & \text{if } i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Algorithm 4 SATURATE

Require: Ground set $\mathcal{V}$, cardinality constraint K, submodular functions $F(\cdot) = [F_1(\cdot), \dots, F_d(\cdot)]$, parameter $\alpha = 1$

$c_{\min} \leftarrow 0$
$c_{\max} \leftarrow \min_i F(\mathcal{V})$
$\hat{S}_K \leftarrow \emptyset$
while $c_{\max} - c_{\min} \geq 1/d$ **do**
    $c \leftarrow (c_{\max} + c_{\min})/2$
    $\hat{S} \leftarrow 0$
    Define $F_c : S \mapsto (1/d) \sum_{i=1}^{d} \min(F_i(S), c)$
    while $F_c(\hat{S}) < c$ **do**
        $\hat{e} \leftarrow \arg\max_{e \in \mathcal{V}/\hat{S}} F_c(\hat{S} \cup \{e\})$
        $\hat{S} \leftarrow \hat{S} \cup \{\hat{e}\}$
    end while
    if $|\hat{S}| > \alpha K$ **then**
        $c_{\max} \leftarrow c$
    else
        $c_{\min} \leftarrow c$
        $\hat{S}_K \leftarrow \hat{S}$
    end if
end while
return $\hat{S}_K$

for the second element e_2, we have:

$$F_i(\{e_2\}) = \begin{cases} C & \text{if } i \in \{2, \dots, i^\star\} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and for the remaining elements $e \in \{e_3, \dots, e_{|\mathcal{V}|}\}$, we have:

$$F_i(\{e\}) = 1 \text{ for all } i \in \{1, \dots, d\}$$

and $F_i(\emptyset) = 0$ for all $i \in \{1, \dots, d\}$. The values of the functions are summarized in Table 1. Clearly, $Q_p(F(\{e_1\})) = Q_p(F(\{e_2\})) = 0$ and $Q_p(F(\{e\})) = 1$ for all the remaining elements $e \in \{e_3, \dots, e_K\}$. Hence, the greedy algorithm will start by creating a set $\hat{S}_1 = \{e\}$ with $e \in \{e_3, \dots, e_{|\mathcal{V}|}\}$. Now, as for any set $S \subseteq \mathcal{V}$ such that $e_1 \notin S$ and $e_2 \notin S$, we have $Q_p(F(S \cup \{e_1\})) = Q_p(F(S \cup \{e_2\})) = Q_p(F(S))$ and $Q_p(F(S \cup \{e\})) = 1 + Q_p(F(S \cup \{e\}))$ for any $e \in \{e_3, \dots, e_K\}/S$, then the greedy algorithm will never add the elements e_1 and e_2 to its set $\hat{S}$. Hence, the output of the greedy algorithm $\hat{S}_K$ will only contain K elements from $\{e_3, \dots, e_{|\mathcal{V}|}\}$ and we have $Q_p(F(\hat{S}_K)) = K$. Finally, since

$$\max_{|S| \leq K} Q_p(F(S)) \geq Q_p(F(\{e_1, e_2\})) = C,$$

setting $C = K/\varepsilon$, it follows that

$$Q_p(F(\hat{S}_K)) = \varepsilon \cdot \left(\frac{K}{\epsilon}\right) \leq \varepsilon \times \max_{|S| \leq K} Q_p(F(S))$$

	F_1	F_2	$\cdots$	$F_{i^\star}$	$F_{i^\star+1}$	$\cdots$	F_d
$\emptyset$	0	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	0
$\{e_1\}$	C	0	$\cdots$	$\cdots$	$\cdots$	$\cdots$	0
$\{e_2\}$	0	C	$\cdots$	C	0	$\cdots$	0
$\{e_3\}$	1	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	1
$\vdots$	$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\vdots$
$\{e_{|\mathcal{V}|}\}$	1	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	1

Table 1. Values of the collection of functions $F_1, \dots, F_d$ designed to prove the result with $C > 1$

which proves the result.

Proof of Proposition 2.2. The proof is a direct extension of Theorem 3 in (Krause et al., 2008). More specifically, for any value of $d \geq 2$, if $p \in [0, 1/d]$, then $Q_p(F(S)) = \min_{i=1\ldots d} F_i(S)$ and the result directly follows from Theorem 3 in (Krause et al., 2008). Now, if $p \in (1/d, 1-1/d)$, set $d^\star = \lceil (1-p)d \rceil$ and consider any collection $F_1, \ldots, F_{d^\star}$ of $d^\star$ non-negative, submodular, monotone functions and define the extended collection of $d > d^\star$ functions $F_1, \ldots, F_d$ where $F_{d^\star+1}(S) = \cdots = F_d(S) = 0$ for any set $S \subseteq \mathcal{V}$. Now, since the functions $F_{d^\star+1}(S) = \cdots = F_d(S) = 0$, it follows that $Q_p(F(S)) = \min_{i=1\ldots d^\star} F_i(S)$. Hence, if there exists an algorithm, which returns a set $\hat{S}$, in polynomial time with regards to the problem instance n, satisfying $Q_p(F(\hat{S}) \geq \gamma(n) \max_{|S| \leq K} Q_p(F(S))$, it follows that the algorithm can return a set $\hat{S}$ such that $\min_{i=1\ldots d^\star} F_i(\hat{S}) \geq \gamma(n) \min_{|S|\leq K} \min_{i=1\ldots d^\star} F_i(S)$ for any collection $F_1, \ldots, F_{d^\star}$ of $d^\star$ functions in polynomial time which implies that P = NP according to Theorem 3 in (Krause et al., 2008) and completes the proof.

Proof of Proposition 3.2. First, observe that since $\lim_{s\to 0} \phi_s(x) = \lim_{s \to 0} \phi_s^{-1}(x) = x$ where $\phi_(x) = (e^{sx}-1)/s$ and $\phi_s^{-1}(x) = \ln(1+sy)/s$ for any $x \in \mathbb{R}$, we know by l'Hôpital rule that both $s \mapsto \phi_s(x)$ and $s \mapsto \phi^{-1}(s)$ are continuous over $\mathbb{R}$. Thus, since $\mu_s(x) = \phi_s^{-1}(\sum_{i=1}^d \phi_s(x_i)/d)$ where the functions $s \mapsto \phi_s(x) = (e^{sx}-1)/s$ and $s \mapsto \phi_s^{-1}(y) = \ln(1+sy)/s$ are continuous over their domain and at $s = 0$ then $s \mapsto \mu_s(x)$ is continuous and we obtain that $\mu_s(x) \to \sum_{i=1}^d x_i/d$ when $s \to 0$. Now, considering $s > 0$, observe that since $\mu_s(x) = (1/s)\ln((1/d)\sum_{i=1}^d e^{sx_i}) = \ln\left(\left((1/d)\sum_{i=1}^d e^{sx_i}\right)^{1/s}\right)$, then $\mu_s(x) = \ln\left((\mathbb{E}\left[|e^X|^s\right])^{1/s}\right)$ where X denotes the random variable taking value x_i with probability $1/d$ for $i \in \{1, \ldots, d\}$. Hence, we have $\mu_s(x) = \ln \|e^X\|_s$ where $\|\cdot\|_s$ denotes the standard L_p-norm. Since L_p-norms are strictly increasing over $\mathbb{R}_+^d$ (by Jensen's inequality), it directly follows that the function $s \mapsto \mu_s(x)$ is non-decreasing over $\mathbb{R}^+$ and since $\|e^x\|_s \to \max_{i=1\ldots d} e^{x_i}$ when $s \to +\infty$, we obtain that $\mu_s(x) \mapsto \max_{i=1\ldots d} x_i$. Finally, using the fact that $\mu_s(x) = -\mu_{-s}(-x)$ and $\max(-x) = -\min(x)$, we obtain the remaining result as a direct consequence of the previous remarks when $s < 0$.

Proof of Proposition 3.3. First, define the random variable $I \sim \mathcal{U}(\{1, \ldots, d\})$ which returns an index uniformly distributed over the dimensions of x and let x_I be the random variable that returns uniformly a component of x. Now, observing that (1) $Q_p(x) = Q_p(x_I)$, (2) the function $x \mapsto (e^{sx}-1)/s$ is positive and non-decreasing and (3) applying Markov's inequality, we obtain that

$$\begin{aligned}
1 - p &\leq \mathbb{P}(x_I \geq Q_p(x_I)) \\
&= \mathbb{P}(x_I \geq Q_p(x)) \\
&= \mathbb{P}\left((e^{sx_I}-1)/s \geq (e^{sQ_p(x)}-1)/s\right) \\
&\leq \frac{\mathbb{E}[(e^{sx_I}-1)/s]}{(e^{sQ_p(x)}-1)/s} \\
&= \frac{(1/d\sum_{i=1}^d e^{sx_i}-1)/s}{(e^{sQ_p(x)}-1)/s}.
\end{aligned}$$

Finally, noticing that $(1/d)\sum_{i=1}^d e^{sx_i} = e^{s\mu_s(x)}$, we obtain that

$$\frac{e^{sQ_p(x)}-1}{s} \leq \frac{e^{s\mu_s(x)}-1}{s(1-p)},$$

and carefully rearranging the previous equation depending on the sign of s leads to the desired result.

Proof of Proposition 3.4. Fix any $s < s'$ and let $S_s \in \arg\max_{|S|\leq K} \mu_s(F(S))$, then, using the fact that for any $x \in \mathbb{R}_+^d$, the function $s \mapsto \mu_s(x)$ is continuous and non-decreasing, we have that $\max_{|S|\leq K} \mu_s(F(S)) = \mu_s(F(S_s)) \leq \mu_{s'}(F(S_s)) \leq \max_{|S|\leq K} \mu_{s'}(F(S))$ proving the first part of the result. Finally, using the fact that for all $S \in \{S \in \mathcal{V} : |S| \leq K\}$ the function $s \mapsto \mu_s(F(S))$ is continuous and that $\lim_{s\to-\infty} \mu_s(F(S)) = \min_{i=1\ldots d} F_i(S)$ and $\lim_{s\to 0}\mu_s(F(S)) = \sum_{i=1}^d F_i(S)/d$ gives the second part of the result.

Proof of Proposition 3.5. To prove the result we use the fact that $\mu_s(F(S)) = g(h(S))$ where $g(y) = \phi_s^{-1}(y) = \frac{1}{s}\ln(1+sy)$ is non-negative and increasing for $s < 0$ and $h(S) = \frac{1}{d}\sum_{i=1}^d \frac{e^{sF_i(S)}-1}{s}$. Now, to prove that the set function $h(S)$ is submodular, fix any sets $A \subseteq B \subseteq \mathcal{V}$ and pick any element $e \in \mathcal{V}/B$, any index $i \in \{1, \ldots, d\}$ and consider any

$s < 0$. By submodularity of the function F_i, we have that:

$$F_i(A \cup \{e\}) - F_i(A) \geq F_i(B \cup \{e\}) - F_i(B).$$

Hence, as the function $x \mapsto 1 - e^{sx}$ is non-decreasing, it follows that:

$$1 - e^{s(F_i(A \cup \{e\}) - F_i(A))} \geq 1 - e^{s(F_i(B \cup \{e\}) - F_i(B))}$$

Observe now that by monotonicity of F_i we have $e^{sF_i(A)} \geq e^{sF_i(B)}$ since $A \subseteq B$. Thus, respectively multiplying the non-negative left-hand (resp. right-hand) side of the previous equation by $e^{sF_i(A)}$ (resp. $e^{sF_i(B)}$), we obtain that:

$$e^{sF_i(A)} - e^{sF_i(A \cup \{e\})} \geq e^{sF_i(B)} - e^{sF_i(B \cup \{e\})},$$

which proves that the function $S \mapsto -e^{sF_i(S)}$ is submodular. Now, since the sum of submodular functions is submodular and the constant function $S \mapsto d$ is also submodular, summing the previous inequality over each component $i \in \{1, \ldots, d\}$ and adding d proves that the function $S \mapsto d - \sum_{i=1}^d e^{sF_i(S)} = \sum_{i=1}^d (1 - e^{sF_i(S)})$ is submodular. Finally, since (1) $\sum_{i=1}^d (1 - e^{sF_i(S)}) \geq 0$ as the functions F_i are non-negative and (2) multiplying a submodular function by a positive constant $d|s| > 0$ preserves submodularity, we know that $S \mapsto (1/d) \sum_{i=1}^d (1 - e^{sF_i(S)})/|s| = (1/d) \sum_{i=1}^d (e^{sF_i(S)} - 1)/s$ which proves the result.

Proof of Proposition 4.3. First, by virtue of Proposition 3.5, we know that $\mu_s(F(S)) = \phi_s^{-1}\left((\sum_{i=1}^d e^{sF_i(S)}/d - 1)/s\right)$ where the function $y \mapsto \phi_s^{-1}(y)$ is strictly increasing over its domain. Hence, the set $\hat{S}_K$ returned by the SOFTSAT algorithm is the same as the one returned by the greedy algorithm run over the function $S \mapsto (1/d \sum_{i=1}^d e^{sF_i(S)} - 1)/s$. Now, as this function is non-negative, monotone and submodular (see Proposition 3.5), it follows from the standard result of the greedy algorithm that the set $\hat{S}_K$ satisfies:

$$\left(1 - \frac{1}{d}\sum_{i=1}^d e^{sF_i(\hat{S}_K)}\right) \geq \left(1 - \frac{1}{e}\right) \cdot \left(1 - \frac{1}{d}\sum_{i=1}^d e^{sF_i(S^\star)}\right)$$

where $S^\star \in \arg\max_{|S| \leq K} \mu_s(F(S))$, which can be re-written as follows:

$$\frac{1}{d}\sum_{i=1}^d e^{sF_i(\hat{S}_K)} \leq 1 - \left(1 - \frac{1}{e}\right)\left(1 - \frac{1}{d}\sum_{i=1}^d e^{sF_i(S^\star)}\right).$$

Now, simply using the fact that $(1/d)\sum_{i=1}^d e^{sF_i(\hat{S}_K)} = e^{s\mu_s(F(\hat{S}_K))}$ and that $(1/d)\sum_{i=1}^d e^{sF_i(S^\star)} = e^{s\mu_s^\star}$ where $\mu_s^\star = \mu_s(F(S^\star))$, the previous inequality can be written as:

$$e^{s\mu_s(F(\hat{S}_K))} \leq 1 - \left(1 - \frac{1}{e}\right)\left(1 - e^{s\mu_s^\star}\right)$$

which gives:

$$\mu_s(F(\hat{S}_K)) \geq \frac{1}{s}\ln\left(1 - \left(1 - \frac{1}{e}\right)(1 - e^{s\mu_s^\star})\right)$$

Thus, using the fact that $\mu_s^\star = -1/s \ln(e^{-s\mu_s^\star})$ and the previous inequality, we obtain that:

$$\begin{aligned}
\mu_s^\star - \mu_s(F(\hat{S}_K)) &= -\frac{1}{s}\ln(e^{-s\mu_s^\star}) - \mu_s(F(\hat{S}_K)) \\
&\leq -\frac{1}{s}\ln\left(e^{-s\mu_s^\star} - \left(1 - \frac{1}{e}\right)(e^{-s\mu_s^\star} - 1)\right) \\
&= -\frac{1}{s}\ln\left(1/e e^{-s\mu_s^\star} + (1 - 1/e)\right) \qquad (4)
\end{aligned}$$

and dividing both sides by $\mu_s^\star$ gives us:

$$\frac{\mu_s^\star - \mu_s(F(\hat{S}_K))}{\mu_s^\star} \leq -\frac{1}{s\mu_s^\star}\log\left(1/e e^{-s\mu_s^\star} + (1 - 1/e)\right).$$

Finally, by Proposition 3.4, we know that $\mu_s^\star \leq \bar{\mu} = \max_{|S|\leq K} \sum_{i=1}^d F_i(S)/d$ which translates to the fact that $s\mu_s^\star \geq s\bar{\mu}$ since $s < 0$. Now observing that the function $x \in \mathbb{R}^- \mapsto -1/x \log(e^{-x}/e + (1 - 1/e))$ is non-increasing, we thus obtain:

$$\begin{aligned} \frac{\mu_s^\star - \mu_s(F(S))}{\mu_s^\star} &\leq -\frac{1}{s\bar{\mu}} \log\left(1/ee^{-s\bar{\mu}} + (1 - 1/e)\right) \\ &= \frac{1}{f(s)} \end{aligned} \tag{5}$$

where $f(s) = -s\bar{\mu}/\log\left(1/ee^{-s\bar{\mu}} + (1 - 1/e)\right)$ and re-arranging the previous equation gives:

$$\mu_s(F(\hat{S}_K)) \geq \left(1 - \frac{1}{f(s)}\right)\mu_s^\star$$

which proves the result.

Proof of Corollary 4.4. Using Proposition 4.3 with s set to $s^\star$ and using the fact that $\max_{|S|\leq K} \mu_{s^\star}(F(S)) \geq \mu_{s^\star}(F(S_p^\star)) = \max_{|S|\leq K} Q_p(F(S))$ directly gives that:

$$\mu_{s^\star}(F(\hat{S}_K)) \geq \left(1 - \frac{1}{f(s^\star)}\right) Q_p(F(S_p^\star)).$$

Now, using the fact that there exists $s' \in \mathbb{R}$ such that $Q_p(F(\hat{S}_K)) = \mu_{s'}(F(\hat{S}_K))$ and the fact that $\mu_{s^\star}(F(\hat{S}_K)) = \mu_{s^\star}(F(\hat{S}_K)) \times \mu_{s'}(F(\hat{S}_K))/\mu_{s'}(F(\hat{S}_K))$, we obtain that:

$$Q_p(F(\hat{S}_K)) \geq \frac{\mu_{s'}(F(\hat{S}_K))}{\mu_{s^\star}(F(\hat{S}_K))}\left(1 - \frac{1}{f(s^\star)}\right) Q_p(F(S_p^\star)).$$

Finally, using the fact that $\mu_{s'}(F(\hat{S}_K)) \geq \min_{i=1\ldots d} F_i(\hat{S}_K)$ and that $\mu_{s^\star}(F(\hat{S}_K)) \leq \max_{i=1\ldots d} F_i(\hat{S}_K)$ gives the result.

Proof of Proposition 4.5. Set $K' = K\lceil\ln(4d)\rceil$, let $s = -\ln(4d)/Q_p(F(S^\star))$, let $\hat{S}_{K'}$ be the output of SOFTSAT tuned with K' and s and set the notations $\hat{Q} = Q_p(F(\hat{S}_{K'}))$ and $Q^\star = Q_p(F(S_K^\star))$. First, observe that for any non-negative (ranked) vector $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ with $x_i \leq x_{i+1}$ for all $i \geq 1$, we have $Q_p(x) = x_{\lceil dp\rceil}$ and:

$$(1)\ \ (d - \lceil dp\rceil + 1)\phi_s(Q_p(x)) \leq \sum_{i=1}^d \phi_s(x_i)$$

and

$$(2)\ \ \sum_{i=1}^d \phi_s(x_i) \leq \lceil dp\rceil\phi_s(Q_p(x)) + (d - \lceil dp\rceil)\phi_s(+\infty).$$

Now using Proposition 3.5, the standard result for the greedy algorithm (Krause & Golovin, 2014) and the fact that $K'/K = \lceil\ln(4d)\rceil \geq \ln(4d)$, we have

$$\begin{aligned} \sum_{i=1}^d \phi_s(F_i(\hat{S}_{K'})) &\geq \left(1 - e^{-K'/K}\right) \cdot \max_{|S|\leq K} \sum_{i=1}^d \phi_s(F_i(S)) \\ &\geq \left(1 - e^{-K'/K}\right) \cdot \sum_{i=1}^d \phi_s(F_i(S_K^\star)) \\ &\geq \left(1 - \frac{1}{4d}\right) \cdot \sum_{i=1}^d \phi_s(F_i(S_K^\star)) \end{aligned}$$

Now, using the fact that $\phi_s(+\infty) = 1/|s|$ and combining (1) and (2) with the previous equation, we obtain that:

$$\lceil dp\rceil\phi_s(\hat{Q}) \geq (1 - \frac{1}{4d})(d - \lceil dp\rceil + 1)\phi_s(Q^\star) - \frac{(d - \lceil dp\rceil)}{|s|}$$

Now using the fact that $\phi_s^{-1}(y) = -1/|s| \ln(1 - |s|y)$ and plugging the value of $s = -\ln(4d)/Q^\star$, we obtain:

$$\begin{aligned}\hat{Q} &\geq -\frac{Q^\star}{\ln(4d)} \ln\left(\frac{d - (d - \lceil pd \rceil + 1)(1 - 1/4d)^2}{\lceil pd \rceil}\right) \\ &= -\frac{Q^\star}{\ln(4d)} \ln\left(1 - \frac{1}{\lceil pd \rceil} + (d - \lceil pd \rceil + 1)(\frac{1}{2d} - \frac{1}{(4d)^2})\right)\end{aligned}$$

Finally, using the fact that $\frac{1}{\lceil pd \rceil} - (d - \lceil pd \rceil + 1)(\frac{1}{2d} - \frac{1}{16d^2}) \geq 1/(2\lceil pd \rceil)$ and the fact that, by convexity of $x \mapsto -\ln(1-x)$, we have $-\ln(1 - ax) \geq -\ln(1 - a)x$ for $a, x \in [0, 1]$, we obtain that:

$$\hat{Q} \geq \frac{\ln(2)}{\lceil pd \rceil \ln(4d)} Q^\star .$$