Robust Submodular Observation Selection

Andreas Krause
SCS, CMU

H. Brendan McMahan
Google, Inc.

Carlos Guestrin
SCS, CMU

Anupam Gupta
SCS, CMU

January 2008
CMU-ML-08-100

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Keywords: Gaussian Processes; Experimental Design; Active Learning; Submodular Functions; Observation Selection

Abstract

In many applications, one has to actively select among a set of expensive observations before making an informed decision. For example, in environmental monitoring, we want to select locations to measure in order to most effectively predict spatial phenomena. Often, we want to select observations which are robust against a number of possible objective functions. Examples include minimizing the maximum posterior variance in Gaussian Process regression, robust experimental design, and sensor placement for outbreak detection. In this paper, we present the *Submodular Saturation* algorithm, a simple and efficient algorithm with strong theoretical approximation guarantees for cases where the possible objective functions exhibit *submodularity*, an intuitive diminishing returns property. Moreover, we prove that better approximation algorithms do not exist unless NP-complete problems admit efficient algorithms. We show how our algorithm can be extended to handle complex cost functions (incorporating non-unit observation cost or communication and path costs). We also show how the algorithm can be used to near-optimally trade off expected-case (e.g., the Mean Square Prediction Error in Gaussian Process regression) and worst-case (e.g., maximum predictive variance) performance. We show that many important machine learning problems fit our robust submodular observation selection formalism, and provide extensive empirical evaluation on several real-world problems. For Gaussian Process regression, our algorithm compares favorably with state-of-the-art heuristics described in the geostatistics literature, while being simpler, faster and providing theoretical guarantees. For robust experimental design, our algorithm performs favorably compared to SDP-based algorithms.

1 Introduction

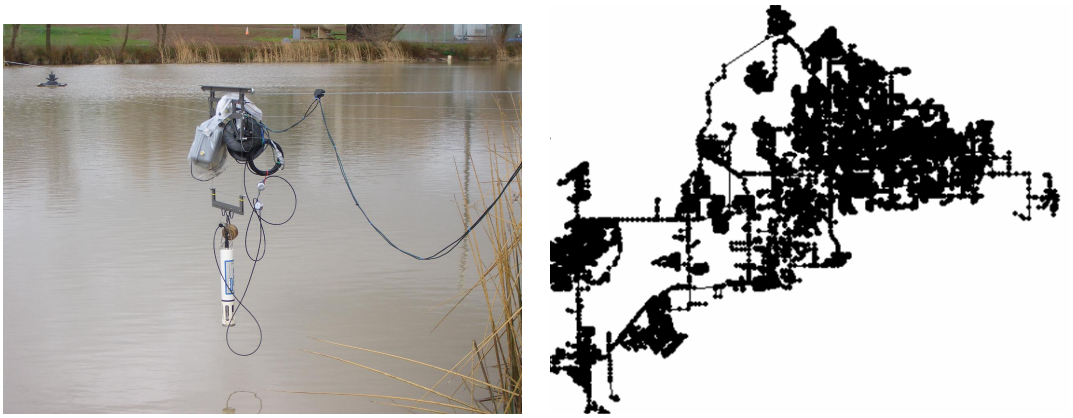
In tasks such as sensor placement for environmental monitoring or experimental design, one has to select among a large set of possible, but expensive, observations. In environmental monitoring, we can choose locations where measurements of a spatial phenomenon (such acidicity in rivers and lakes, *cf.,* Figure 1(a)) should be obtained. In experimental design, we frequently have a menu of possible experiments which can be performed. Often, there are several different objective functions which we want to simultaneously optimize. For example, in the environmental monitoring problem, we want to minimize the marginal posterior variance of our acidicity estimate at all locations simultaneously. In experimental design, we often have uncertainty about the model parameters, and we want our experiments to be informative no matter what the true parameters of the model are. In sensor placement for contamination detection in water distribution networks (*cf.,* Figure 1(b)), we want to place sensors in order to quickly detect any possible contamination event.

Our goal in all these problems is to select observations (sensor locations, experiments) which are *robust* against a worst-case objective function (location to evaluate predictive variance, model parameters, contamination event, etc.). Often, the individual objective functions, e.g., the marginal variance at one location, or the information gain for a fixed set of parameters (Das and Kempe, 2007; Krause et al., 2007b; Krause and Guestrin, 2005; Guestrin et al., 2005), satisfy *submodularity*, an intuitive diminishing returns property: Adding a new observation helps less if we have already made many observations, and more if we have made few observation thus far. While NP-hard, the problem of selecting an optimal set of k observations maximizing a single submodular objective can be approximately solved using a simple greedy forward-selection algorithm, which is guaranteed to perform near-optimally (Nemhauser et al., 1978). However, as we show, this simple *myopic* algorithm performs arbitrarily badly in the case of a worst-case objective function. In this paper, we address the fundamental problem of nonmyopically selecting observations which are robust against such an adversarially chosen submodular objective function. In particular:

- We present SATURATE, an efficient algorithm for the robust submodular observation selection problem. Our algorithm guarantees solutions which are at least as informative as the optimal solution, at only a slightly higher cost.
- We prove that our approximation guarantee is the best possible, i.e., the guarantee cannot be improved unless NP-complete problems admit efficient algorithms.
- We discuss several extensions of our approach, handling complex cost functions and trading off worst-case and average-case performance.
- We extensively evaluate our algorithm on several real-world tasks, including minimizing the maximum posterior variance in Gaussian Process regression, finding experiment designs which are robust with respect to parameter uncertainty, and sensor placement for outbreak detection.

This manuscript is organized as follows. In Section 2, we formulate the robust submodular observation selection problem, and in Section 3, we analyze its hardness. We subsequently present SATURATE, an efficient approximation algorithm for this problem (Section 4), and show that our approximation guarantees are best possible, unless NP-complete

problems admit efficient algorithms (Section 5). In Section 6, we discuss how many important machine learning problems are instances of our robust submodular observation selection formalism. We then discuss extensions (Section 7) and evaluate the performance of SATURATE on several real-world observation selection problems (Section 8). Section 9 presents heuristics to improve the computational performance of our algorithm, Section 10 reviews related work, and Section 11 presents our conclusions.



(a) *NIMS deployed at UC Merced* (b) *Water distribution network*

Figure 1: (a) Deployment of the Networked Infomechanical System (NIMS, Harmon et al. 2006) to monitor a lake near UC Merced. (b) Illustration of the municipal water distribution network considered in the Battle of the Water Sensor Networks challenge (*cf.,* Ostfeld et al., 2008).

2 Robust Submodular Observation Selection

In this section, we first review the concept of submodularity (Section 2.1), and then introduce the *robust submodular observation selection (*RSOS*)* problem (Section 2.2).

2.1 Submodular Observation Selection

Let us consider a spatial prediction problem, where we want to estimate the pH values across a horizontal transect of a river, e.g., using the NIMS robot shown in Figure 1(a). We can discretize the space into a finite number of locations $\mathcal{V}$, where we can obtain measurements, and model a joint distribution $P(\mathcal{X}_\mathcal{V})$ over variables $\mathcal{X}_\mathcal{V}$ associated with these locations. One example of such models, which have found common use in geostatistics (*cf.,* Cressie, 1991), are Gaussian Processes (*cf.,* Rasmussen and Williams, 2006). Based on such a model, a typical goal in spatial monitoring is to select a subset of locations $\mathcal{A} \subseteq \mathcal{V}$ to observe, such that the average predictive variance,

$$V(\mathcal{A}) = \frac{1}{n} \sum_i \sigma^2_{i|\mathcal{A}},$$

is minimized (*cf.,* Section 6.1 for more details). Hereby, $\sigma^2_{i|\mathcal{A}}$ denotes the predictive variance at location i after observing locations $\mathcal{A}$, i.e.,

$$\sigma^2_{i|\mathcal{A}} = \int P(\mathbf{x}_\mathcal{A}) \mathbb{E}\left[(\mathcal{X}_i - \mathbb{E}\left[\mathcal{X}_i \mid \mathbf{x}_\mathcal{A}\right])^2 \mid \mathbf{x}_\mathcal{A} \right] d\mathbf{x}_\mathcal{A}.$$

Unfortunately, the problem

$$\mathcal{A}^* = \operatorname*{argmin}_{|\mathcal{A}| \leq k} V(\mathcal{A})$$

is NP-hard in general (Das and Kempe, 2007), and the number of candidate solutions is very large, so generally we cannot expect to efficiently find the optimal solution. Fortunately, as Das and Kempe (2007) show, in many cases, the *variance reduction*

$$F_s(\mathcal{A}) = \sigma^2_s - \sigma^2_{s|\mathcal{A}}$$

at any particular location s, satisfies the following diminishing returns behavior: Adding a new observation reduces the variance at s more, if we have made few observations so far, and less, if we have already made many observations. This formalism can be formalized using the combinatorial concept of *submodularity* (*cf.,* Nemhauser et al., 1978):

Definition 1 *A set function $F : 2^\mathcal{V} \to \mathbb{R}$ is called* submodular*, if for all subsets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ it holds that $F(\mathcal{A} \cup \mathcal{B}) + F(\mathcal{A} \cap \mathcal{B}) \leq F(\mathcal{A}) + F(\mathcal{B})$.*

Nemhauser et al. (1978) prove a convenient characterization of submodular functions: F is submodular if and only if for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \backslash \mathcal{B}$ it holds that $F(\mathcal{A} \cup \{s\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{s\}) - F(\mathcal{B})$. This characterization exactly matches our diminishing returns intuition about the variance reduction F_s at location s. Since each of the variance reduction functions F_s is submodular, the *average variance reduction*

$$F(\mathcal{A}) = V(\emptyset) - V(\mathcal{A}) = \frac{1}{n} \sum_s F_s(\mathcal{A})$$

is also submodular. The average variance reduction is also *monotonic*, i.e., for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ it holds that $F(\mathcal{A}) \leq F(\mathcal{B})$, and *normalized* ($F(\emptyset) = 0$).

Hence, the problem of minimizing the average variance is an instance of the problem

$$\max_{\mathcal{A} \subseteq \mathcal{V}} F(\mathcal{A}), \quad \text{subject to} \quad |\mathcal{A}| \leq k, \tag{2.1}$$

where F is normalized, monotonic and submodular, and k is a bound on the number of observations we can make. As Krause and Guestrin (2007a) show (also, *cf.,* Appendix C), many other observation selection problems are instances of Problem (2.1).

Since solving Problem (2.1) is NP-hard in most interesting instances (Feige, 1998; Krause et al., 2006, 2007b; Das and Kempe, 2007), in practice, heuristics are often used. One such heuristic is the *greedy algorithm*. This algorithm starts with the empty set, and iteratively adds the element $s^* = \operatorname{argmax}_{s \in \mathcal{V} \backslash \mathcal{A}} F(\mathcal{A} \cup \{s\})$, until k elements have been selected. Perhaps surprisingly, a fundamental result by Nemhauser et al. (1978) states that for submodular functions, the greedy algorithm achieves a constant factor approximation:

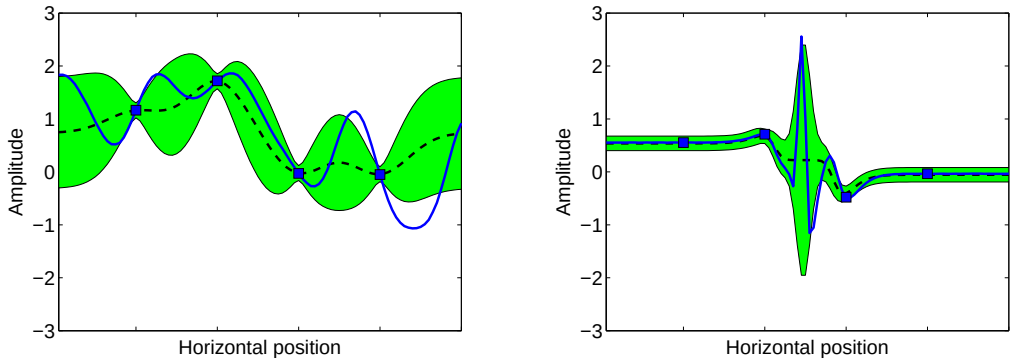
Theorem 2 (Nemhauser et al. 1978) *In the case of any normalized, monotonic submodular function F, the set $\mathcal{A}_G$ obtained by the greedy algorithm achieves at least a constant fraction $(1-1/e)$ of the objective value obtained by the optimal solution, i.e.,*

$$F(\mathcal{A}_G) \geq (1-1/e) \max_{|\mathcal{A}| \leq k} F(\mathcal{A}).$$

Moreover, no polynomial time algorithm can provide a better approximation guarantee unless P = NP (Feige, 1998).

2.2 The Robust Submodular Observation Selection (RSOS) Problem

For phenomena, such as the one indicated in Figure 2(a), which are spatially homogeneous (isotropic), maximizing this average variance reduction leads to effective variance reduction everywhere in the space. However, many spatial phenomena are nonstationary, being smooth in certain areas and highly variable in others, such as the example indicated in Figure 2(b). In such a case, maximizing the average variance reduction will typically put only few examples in the areas highly variable areas. However, those regions are typically the most interesting, since they are most difficult to predict. In such cases, we might want to simultaneously minimize the variance everywhere in the space.



(a) *High average variance* (b) *High maximum variance*

Figure 2: Spatial predictions using Gaussian Processes with a small number of observations. The blue solid line indicates the unobserved latent function, and blue squares indicate observations. The plots also show confidence bands (green). Dashed line indicates the prediction. (b) shows an example with high maximum predictive variance, but low average variance, whereas (a) shows an example with high average variance, but lower maximum variance. Note, that in (b) we are most uncertain about the most variable (and interesting, since it is hard to predict) part of the curve, suggesting that the maximum variance should be optimized.

More generally, in many applications (such as the spatial monitoring problem discussed above, and several other examples which we present in Section 6), we want to perform equally well with respect to *multiple* objectives. We will hence consider settings where we are given a *collection* of normalized monotonic submodular functions $F_1, \ldots, F_m$, and we want to solve

$$\max_{\mathcal{A} \subseteq \mathcal{V}} \min_i F_i(\mathcal{A}), \quad \text{subject to} \quad |\mathcal{A}| \leq k. \tag{2.2}$$

$\mathcal{A}$	$F_1(\mathcal{A})$	$F_2(\mathcal{A})$	$\min_i F_i(\mathcal{A})$
$\emptyset$	0	0	0
$\{s_1\}$	1	0	0
$\{s_2\}$	0	1	0
$\{t_1\}$	ε	ε	ε
$\{t_2\}$	ε	ε	ε
$\{s_1, s_2\}$	1	1	1
$\{s_1, t_1\}$	$1+\varepsilon$	ε	ε
$\{s_1, t_2\}$	$1+\varepsilon$	ε	ε
$\{s_2, t_1\}$	ε	$1+\varepsilon$	ε
$\{s_2, t_2\}$	ε	$1+\varepsilon$	ε
$\{t_1, t_2\}$	2ε	2ε	2ε

Table 1: Functions F_1 and F_2 used in the counterexample.

The goal of Problem (2.2) is to find a set $\mathcal{A}$ of observations, which is robust against the worst possible objective, $\min_i F_i$, from our set of possible objectives. Consider the spatial monitoring setting for example, and assume that the prior variance σ_i^2 is constant (we will relax this assumption in Section 7.2) over all locations i. Then, the problem of minimizing the maximum variance is equivalent to maximizing the minimum variance reduction, i.e., solving Problem (2.2) where F_i is the variance reduction at location i.

We call Problem (2.2) the *Robust Submodular Observation Selection (*RSOS*)* problem. Note, that even if the F_i are all submodular, $F_{wc}(\mathcal{A}) = \min_i F_i(\mathcal{A})$ is generally *not* submodular. In fact, we show below that, in this setting, the simple greedy algorithm (which performs near-optimally in the single-criterion setting) can perform arbitrarily badly.

3 Hardness of the Robust Submodular Observation Selection Problem

Given the near-optimal performance of the greedy algorithm for the single-objective problem, a natural question is if the performance guarantee generalizes to the more complex robust optimization setting. Unfortunately, this hope is far from true, even in the simpler case of *modular* (additive) functions F_i. Consider a case with two submodular functions, F_1 and F_2, where the set of observations is $\mathcal{V} = \{s_1, s_2, t_1, t_2\}$. The functions take values as indicated in Table 1. Optimizing for a set of 2 elements, the greedy algorithm maximizing $F_{wc}(\mathcal{A}) = \min\{F_1(\mathcal{A}), F_2(\mathcal{A})\}$ would first choose t_1 (or t_2), as this choice increases the objective $\min\{F_1, F_2\}$ by ε, as opposed to 0 for s_1 and s_2. The greedy solution for $k = 2$ would then be the set $\{t_1, t_2\}$, obtaining a score of 2ε. However, the optimal solution with $k = 2$ is $\{s_1, s_2\}$, with a score of 1. Hence, as $\varepsilon \to 0$, the greedy algorithm performs arbitrarily worse than the optimal solution.

Given that the greedy algorithm performs arbitrarily badly, our next hope would be to obtain a different good approximation algorithm. However, we can show that most likely this is not possible:

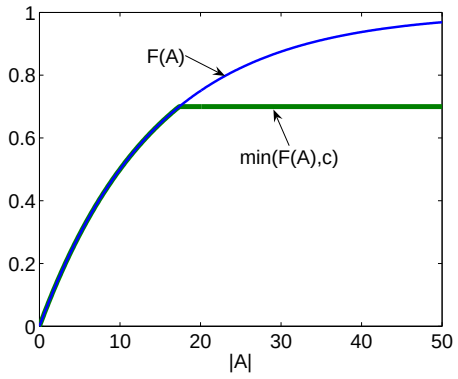


Figure 3: Truncating an objective function F preserves submodularity and monotonicity.

Theorem 3 *Unless* P = NP*, there cannot exist* any *polynomial time approximation algorithm for Problem* (2.2)*. More precisely: Let n be the size of the problem instance, and $\gamma(\cdot) > 0$ be any positive function of n. If there exists a polynomial-time algorithm which is guaranteed to find a set $\mathcal{A}'$ of size k such that $\min_i F_i(\mathcal{A}') \geq \gamma(n) \max_{|\mathcal{A}|\leq k} \min_i F_i(\mathcal{A})$, then* P = NP*.*

Thus, unless P = NP, there cannot exist any algorithm which is guaranteed to provide, e.g., even an exponentially small fraction ($\gamma(n) = 2^{-n}$) of the optimal solution. All proofs can be found in the Appendix.

4 The Submodular Saturation Algorithm

Since Theorem 3 rules out *any* approximation algorithm which respects the constraint k on the size of the set $\mathcal{A}$, our only hope for non-trivial guarantees requires us to relax this constraint. We now present an algorithm that finds a set of observations which perform at least as well as the optimal set, but at slightly increased cost; moreover, we show that no efficient algorithm can provide better guarantees (under reasonable complexity-theoretic assumptions). For now we assume all F_i take only integral values; this assumption is relaxed in Section 7.1. The key idea is to consider the following alternative formulation:

$$\max_{c,\mathcal{A}} c, \qquad \text{subject to} \quad c \leq F_i(\mathcal{A}) \text{ for } 1 \leq i \leq m \text{ and } |\mathcal{A}| \leq \alpha k. \tag{4.1}$$

We want a set $\mathcal{A}$ of size at most αk, such that $F_i(\mathcal{A}) \geq c$ for all i, and c is as large as possible. Hereby, $\alpha \geq 1$ is a parameter relaxing the constraint on $|\mathcal{A}|$. If $\alpha = 1$, we recover the original problem (2.2): In this case, maximizing c subject to the existence of a set $\mathcal{A}$, $|\mathcal{A}| \leq k$ such that $F_i(\mathcal{A}) \geq c$ for all i is equivalent to maximizing $\min_i F_i(\mathcal{A})$. For arbitrary values of $\alpha \geq 1$, we can conceptually solve program (4.1) as follows: For any given value c, we find the cheapest set $\mathcal{A}$ with $F_i(\mathcal{A}) \geq c$ for all i. If this cheapest set has at most αk elements, then $(c, \mathcal{A})$ is feasible. A binary search on c would then allow us to find the optimal solution with the maximum feasible c.

We first show how to *approximately* solve Equation (4.1) for a fixed c. For $c > 0$ define $\widehat{F}_{i,c}(\mathcal{A}) = \min\{F_i(\mathcal{A}), c\}$, the original function F_i truncated at score level c. These $\widehat{F}_{i,c}$

GPC ($\overline{F}_c$, c)
$\mathcal{A} \leftarrow \emptyset$;
while $\overline{F}_c(\mathcal{A}) < c$ **do**
  foreach $s \in \mathcal{V} \setminus \mathcal{A}$ **do** $\delta_s \leftarrow \overline{F}_c(\mathcal{A} \cup \{s\}) - \overline{F}_c(\mathcal{A})$;
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\text{argmax}_s\, \delta_s\}$;
end

Algorithm 1: The greedy submodular partial cover (GPC) algorithm.

functions are also submodular (Fujito, 2000). Figure 3 illustrates this truncation concept. Let $\overline{F}_c(\mathcal{A}) = \frac{1}{m}\sum_i \widehat{F}_{i,c}(\mathcal{A})$ be their average value. Submodular functions are closed under convex combinations, so $\overline{F}_c$ is submodular and monotonic. Furthermore, $F_i(\mathcal{A}) \geq c$ for all $1 \leq i \leq m$ if and only if $\overline{F}_c(\mathcal{A}) = c$. Hence, in order to determine whether some c is feasible, we need to find the smallest set such that $\overline{F}_c(\mathcal{A}) = c = \overline{F}_c(\mathcal{V})$, i.e., solve:

$$\mathcal{A}_c = \operatorname*{argmin}_{\mathcal{A} \subseteq \mathcal{V}} |\mathcal{A}|, \quad \text{such that} \quad \overline{F}_c(\mathcal{A}) = c. \tag{4.2}$$

Problems of the form $\min_{\mathcal{A}} |\mathcal{A}|$ such that $F(\mathcal{A}) = F(\mathcal{V})$, where F is a normalized monotonic submodular function, are called *submodular covering problems*. Since $\overline{F}_c$ satisfies these requirements, (4.2) is an instance of such a submodular covering problem. While such problems are NP-hard in general (Feige, 1998), Wolsey (1982) shows that the greedy algorithm (*cf.,* Algorithm 1) achieves near-optimal performance on this problem. Using his result, we find:

Lemma 4 *Given monotonic submodular functions $F_1, \ldots, F_m$ and a (feasible) constant c, Algorithm 1 (with input $\overline{F}_c$) finds a set $\mathcal{A}_G$ such that $F_i(\mathcal{A}_G) \geq c$ for all i, and $|\mathcal{A}_G| \leq \alpha|\mathcal{A}^*|$, where $\mathcal{A}^*$ is an optimal solution, and*[1]

$$\alpha = 1 + \log\left(\max_{s \in \mathcal{V}} \sum_i F_i(s)\right) \geq 1 + \log\left(m \max_{s \in \mathcal{V}} \overline{F}_c(s)\right).$$

We can compute this approximation guarantee α for any given instance of the RSOS problem. Hence, if for a given value of c the greedy algorithm returns a set of size greater than αk, there cannot exist a solution $\mathcal{A}'$ with $|\mathcal{A}'| \leq k$ with $F_i(\mathcal{A}') \geq c$ for all i; thus, the optimal solution to the RSOS problem must be less than c. We can use this argument to conduct a binary search to find the optimal value of c. We call Algorithm 2, which formalizes this procedure, the *submodular saturation algorithm* (SATURATE), as the algorithm considers the truncated objectives $\widehat{F}_{i,c}$, and chooses sets which *saturate* all these objectives. Theorem 5 (given below) states that SATURATE is guaranteed to find a set which achieves worst-case score $\min_i F_i$ at least as high as the optimal solution, if we allow the set to be logarithmically larger than the optimal solution.

Theorem 5 *For any integer k,* SATURATE *finds a solution $\mathcal{A}_S$ such that*

$$\min_i F_i(\mathcal{A}_S) \geq \max_{|\mathcal{A}| \leq k} \min_i F_i(\mathcal{A}) \quad \text{and} \quad |\mathcal{A}_S| \leq \alpha k,$$

[1]This bound is only meaningful for integral F_i, otherwise it could be arbitrarily improved by scaling the F_i. We relax the constraint on integrality of the F_i in Section 7.1.

SATURATE $(F_1, \ldots, F_m, k, \alpha)$
$c_{\min} \leftarrow 0$; $c_{\max} \leftarrow \min_i F_i(\mathcal{V})$; $\mathcal{A}_{best} \leftarrow \emptyset$;
while $(c_{\max} - c_{\min}) \geq \frac{1}{m}$ **do**
  $c \leftarrow (c_{\min} + c_{\max})/2$;
  Define $\overline{F}_c(\mathcal{A}) \leftarrow \frac{1}{m} \sum_i \min\{F_i(\mathcal{A}), c\}$;
  $\widehat{\mathcal{A}} \leftarrow GPC(\overline{F}_c, c)$;
  if $|\widehat{\mathcal{A}}| > \alpha k$ **then**
    $c_{\max} \leftarrow c$;
  else
    $c_{\min} \leftarrow c$; $\mathcal{A}_{best} = \widehat{\mathcal{A}}$
  end
end

Algorithm 2: The Submodular Saturation algorithm.

for $\alpha = 1 + \log(\max_{s \in \mathcal{V}} \sum_i F_i(s))$. *The total number of submodular function evaluations is*

$$\mathcal{O}\left(|\mathcal{V}|^2 m \log\left(\sum_i F_i(\mathcal{V})\right)\right).$$

Note, that the algorithm still makes sense for any value of α. However, if $\alpha < 1 + \log(\max_{s \in \mathcal{V}} \sum_i F_i(s))$, the guarantee of Theorem 5 does not hold. If we had an exact algorithm for submodular coverage, $\alpha = 1$ would be the correct choice. Since the greedy algorithm solves submodular coverage very effectively, in our experiments, we call SATURATE with $\alpha = 1$, which empirically performs very well.

If we apply SATURATE to the example problem described in Section 3, we would start with $c_{\max} = 1$. Running the coverage algorithm (GPC) with $c = 0.5$ would first pick element s_1 (or s_2), since $\overline{F}_c(\{s_1\}) = 0.5$, and, next, pick s_2 (or s_1 resp.), hence finding the optimal solution.

The worst-case running time guarantee is quite pessimistic, and in practice the algorithm is much faster: Using a priority queue and lazy evaluations, Algorithm 1 can be sped up drastically (*cf.,* Robertazzi and Schwartz 1989 for details). Furthermore, in practical implementations, one would stop GPC once $\alpha k + 1$ elements have been selected, which already proves that the optimal solution with k elements cannot achieve score c. Also, Algorithm 2 can be terminated once $c_{\max} - c_{\min}$ is sufficiently small; in our experiments, 10-15 iterations usually sufficed.

5 Hardness of Bicriterion Approximation

Guarantees of the form presented in Theorem 5 are often called *bicriterion* guarantees. Instead of requiring that the obtained objective score is close to the optimal score *and all* constraints are exactly met, a bicriterion guarantee requires a bound on the suboptimality of the objective, as well as bounds on how much the constraints are violated. Theorem 3 showed that – unless P = NP – no approximation guarantees can be obtained which do not violate the constraint on the cost k, thereby necessitating the bricriterion analysis.

One might ask, whether the guarantee on the size of the set, α, can be improved. Unfortunately, this is not likely, as the following result shows:

Theorem 6 *If there were a polynomial time algorithm which, for any integer k, is guaranteed to find a solution $\mathcal{A}_S$ such that $\min_i F_i(\mathcal{A}_S) \geq \max_{|\mathcal{A}|\leq k} \min_i F_i(\mathcal{A})$ and $|\mathcal{A}_S| \leq \beta k$, where $\beta \leq (1-\varepsilon)(1+\log \max_{s\in\mathcal{V}} \sum_i F_i(s))$ for some fixed $\varepsilon > 0$, then* NP $\subseteq$ DTIME($n^{\log\log n}$).

Hereby, DTIME($n^{\log\log n}$) is a class of deterministic, slightly superpolynomial (but subexponential) algorithms (Feige, 1998); the inclusion NP $\subseteq$ DTIME($n^{\log\log n}$) is considered unlikely (Feige, 1998). Taken together, Theorem 3 and Theorem 6, provide strong theoretical evidence that SATURATE achieves best possible theoretical guarantees for the problem of maximizing the minimum over a set of submodular functions.

6 Examples of Robust Submodular Observation Selection problems

We now demonstrate that many important machine learning problems can be phrased as RSOS problems. Section 8 provides more details and experimental results for these domains.

6.1 Minimizing the Maximum Kriging Variance

Consider a Gaussian Process (GP) (*cf.,* Rasmussen and Williams, 2006) $\mathcal{X}_\mathcal{V}$ defined over a finite set of locations (indices) $\mathcal{V}$. Hereby, $\mathcal{X}_\mathcal{V}$ is a set of random variables, one variable $\mathcal{X}_s$ for each location $s \in \mathcal{V}$. Given a set of locations $\mathcal{A} \subseteq \mathcal{V}$ which we observe, we can compute the predictive distribution $P(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A})$, i.e., the distribution of the variables $\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}$ at the unobserved locations $\mathcal{V} \setminus \mathcal{A}$, conditioned on the measurements at the selected locations, $\mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}$. Let $\sigma^2_{s|\mathcal{A}}$ be the residual variance after making observations at $\mathcal{A}$. Let $\Sigma_{\mathcal{A}\mathcal{A}}$ be the covariance matrix of the measurements at the chosen locations $\mathcal{A}$, and $\Sigma_{s\mathcal{A}}$ be the vector of cross-covariances between the measurements at s and $\mathcal{A}$. Then, the predictive variance (often called Kriging variance in the geostatistics literature), given by

$$\sigma^2_{s|\mathcal{A}} = \sigma^2_s - \Sigma_{s\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\Sigma_{\mathcal{A}s},$$

depends only on the set $\mathcal{A}$, and *not* on the observed values $\mathbf{x}_\mathcal{A}$. As argued in Section 2, an often (especially in the case of nonstationary phenomena) appropriate criterion is to select locations $\mathcal{A}$ such that the maximum marginal variance is as small as possible, i.e., we want to select a subset $\mathcal{A}^* \subseteq \mathcal{V}$ of locations to observe such that

$$\mathcal{A}^* = \operatorname*{argmin}_{|\mathcal{A}|\leq k} \max_{s\in\mathcal{V}} \sigma^2_{s|\mathcal{A}}. \tag{6.1}$$

Let us assume for now that the a priori variance σ^2_s is constant for all locations s (in Section 7, we show how our approach generalizes to non-constant marginal variances). Furthermore, let us define the *variance reduction* $F_s(\mathcal{A}) = \sigma^2_s - \sigma^2_{s|\mathcal{A}}$. Solving Problem (6.1) is then equivalent to maximizing the minimum variance reduction over all locations s. For a

particular location s, Das and Kempe (2007) show that the variance reduction F_s (often) is a monotonic submodular function. Hence the problem

$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}|\leq k} \min_{s\in\mathcal{V}} F_s(\mathcal{A}) = \operatorname*{argmax}_{|\mathcal{A}|\leq k} \min_{s\in\mathcal{V}} \sigma_s^2 - \sigma_{s|\mathcal{A}}^2$$

is an instance of the RSOS problem.

6.2 Variable Selection under Parameter Uncertainty

Consider an application, where we want to diagnose a failure of a complex system, by performing a number of tests. We can model this problem by using a set of discrete random variables $\mathcal{X}_\mathcal{V} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ indexed by $\mathcal{V} = \{1, \ldots, n\}$, which model both the hidden state of the system and the outcomes of the diagnostic tests. The interaction between these variables is modeled by a joint distribution $P(\mathcal{X}_\mathcal{V} \mid \theta)$ with parameters θ. Krause et al. (2007b) and Krause and Guestrin (2005) show that many variable selection problems can be formulated as the problem of optimizing a submodular utility function (measuring, e.g., the information gain $I(\mathcal{X}_\mathcal{U}, \mathcal{X}_\mathcal{A})$ with respect to some variables of interest $\mathcal{U}$, or the mutual information $I(\mathcal{X}_\mathcal{A}; \mathcal{X}_{\mathcal{V}\setminus\mathcal{A}})$ between the observed and unobserved variables, etc.). However, the informativeness of a chosen set $\mathcal{A}$ typically depends on the particular parameters θ, and these parameters might be uncertain. In some applications, it might not be reasonable to impose a prior distribution over θ, and we may want to perform well even under the worst-case parameters. In these cases, we can associate, with each parameter setting θ, a different submodular objective function F_θ, for example,

$$F_\theta(\mathcal{A}) = I(\mathcal{X}_\mathcal{A}; \mathcal{X}_\mathcal{U} \mid \theta),$$

and we might want to select a set $\mathcal{A}$ which simultaneously performs well for all possible parameter values. In practice, we can discretize the set of possible parameter values θ (for example around a 95% confidence interval estimated from initial data) and optimize the worst case F_θ over the resulting discrete set of parameters.

6.3 Robust Experimental Designs

Another application is experimental design under nonlinear dynamics (Flaherty et al., 2006). The goal is to estimate a set of parameters θ of a nonlinear function $y = f(\mathbf{x}, \theta) + w$, by providing a set of experimental stimuli $\mathbf{x}$, and measuring the (noisy) response y. In many cases, experimental design for linear models (where $y = A(\mathbf{x})^T\theta + w$ with Gaussian noise w) can be efficiently solved by semidefinite programming (Boyd and Vandenberghe, 2004). In the nonlinear case, a common approach (*cf.,* Chaloner and Verdinelli, 1995) is to *linearize* f around an initial parameter estimate θ_0, i.e.,

$$y = f(\mathbf{x}, \theta_0) + V(\mathbf{x})(\theta - \theta_0) + w, \tag{6.2}$$

where $V(\mathbf{x})$ is the Jacobian of f with respect to the parameters θ, evaluated at θ_0. Subsequently, a *locally-optimal* design is sought, which is optimal for the linear design problem (6.2) for initial parameter estimates θ_0. Flaherty et al. (2006) show that the efficiency of

such a locally optimal design can be very sensitive with respect to the initial parameter estimates θ_0. Consequently, they develop an efficient semi-definite program (SDP) for E-optimal design (i.e., the goal is to minimize the maximum eigenvalue of the error covariance) which is robust against perturbations of the Jacobian V. However, it might be more natural to directly consider robustness with respect to perturbation of the initial parameter estimates θ_0, around which the linearization is performed. We show how to find (Bayesian A-optimal) designs which are robust against uncertainty in these parameter estimates. In this setting, the objectives $F_{\theta_0}(\mathcal{A})$ are the reductions of the trace of the parameter covariance,

$$F_{\theta_0}(\mathcal{A}) = \operatorname{tr}\left(\Sigma_{\theta}^{(\theta_0)}\right) - \operatorname{tr}\left(\Sigma_{\theta|\mathcal{A}}^{(\theta_0)}\right),$$

where $\Sigma^{(\theta_0)}$ is the joint covariance of observations and parameters after linearization around θ_0; thus, F_{θ_0} is the sum of marginal parameter variance reductions, which are (often) individually monotonic and submodular (Das and Kempe, 2007), and so F_{θ_0} is monotonic and submodular as well. Hence, in order to find a robust design, we maximize the minimum variance reduction, where the minimum is taken over (a discretization into a finite subset of) all initial parameter values θ_0.

6.4 Sensor Placement for Outbreak Detection

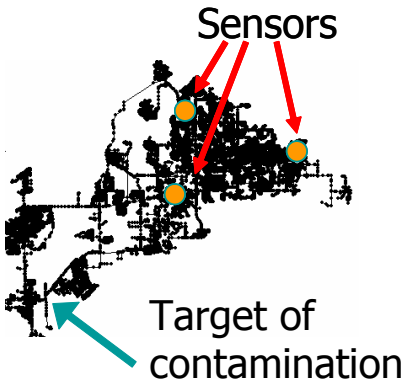


Figure 4: Securing a municipal water distribution network against contaminations performed under knowledge of the sensor placement is another instance of the RSOS problem.

Another class of examples are outbreak detection problems on graphs, such as contamination detection in water distribution networks (Leskovec et al., 2007). Here, we are given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and a phenomenon spreading dynamically over the graph. We define a set of *intrusion scenarios* $\mathcal{I}$; each scenario $i \in \mathcal{I}$ models an outbreak (e.g., spreading of contamination) starting from a given node $s \in \mathcal{V}$ in the network. By placing sensors at a set of locations $\mathcal{A} \subseteq \mathcal{V}$, we can detect such an outbreak, and thereby minimize the adverse effects on the network.

More formally, for each possible outbreak scenario $i \in \mathcal{I}$ and for each node $v \in \mathcal{V}$ we define the detection time $T_i(v)$ as the time when the outbreak affects node v (and $T_i(v) = \infty$ if node v is never affected). We furthermore define a penalty function $\pi_i(t)$ which models

the penalty incurred for detecting outbreak i at time t. We require $\pi_i(t)$ to be monotonically non-decreasing in t (i.e., we never prefer late over early detection), and bounded above by $\pi_i(\infty) \in \mathbb{R}$. Our goal is to minimize the worst-case penalty: We extend π_i to observation sets $\mathcal{A}$ as $\pi_i(\mathcal{A}) = \pi_i\left(\min_{s\in\mathcal{A}} T_i(s)\right)$. Then, our goal is to solve

$$\mathcal{A}^* = \operatorname*{argmin}_{|\mathcal{A}|\leq k} \max_{i\in\mathcal{I}} \pi_i(\mathcal{A}).$$

Equivalently, we can define the *penalty reduction* $F_i(\mathcal{A}) = \pi_i(\infty) - \pi_i(\mathcal{A})$. Clearly, $F_i(\emptyset) = 0$, F_i is monotonic. In Leskovec et al. (2007), it was shown that F_i is also guaranteed to be submodular. For now, let us assume that $\pi_i(\infty)$ is constant for all i (we will relax this assumption in Section 7.2). Our goal in sensor placement is then to select a set of sensors $\mathcal{A}$ such that the minimum penalty reduction is as large as possible, i.e., we want to select

$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}|\leq k} \min_{i\in\mathcal{I}} F_i(\mathcal{A}).$$

In other words, an adversary observes our sensor placement $\mathcal{A}$, and then decides on an intrusion i for which our utility $F_i(\mathcal{A})$ is as small as possible. Hence, our goal is to find a placement $\mathcal{A}$ which performs well against such an adversarial opponent.

6.5 Robustness Against Sensor Failures and Feature Deletion

Another interesting instance of the RSOS problem arises in the context of robust sensor placements. For example, in the outbreak detection problem, sensors might *fail*, due to hardware problems or manipulation by an adversary. We can model this problem in the following way: Consider the case where all sensors at a subset $\mathcal{B} \subseteq \mathcal{V}$ of locations fail. Given a submodular function F (e.g., the utility for placing a set of sensors), and the set $\mathcal{B} \subseteq \mathcal{V}$ of failing sensors, we can define a new function $F_\mathcal{B}(\mathcal{A}) = F(\mathcal{A} \setminus \mathcal{B})$, corresponding to the (reduced) utility of placement $\mathcal{A}$ after the sensor failures. It is easy to show that if F is nondecreasing and submodular, so is $F_\mathcal{B}$. Hence, the problem of optimizing sensor placements which are robust to sensor failures results in a problem of simultaneously maximizing a collection of submodular functions, e.g., for the worst-case failure of $k' < k$ sensors we solve

$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}|\leq k} \min_{|\mathcal{B}|\leq k'} F_\mathcal{B}(\mathcal{A}).$$

We can also combine the optimization against adversarial contamination scenarios as discussed in Section 6.3 with adversarial sensor failures, and optimize

$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}|\leq k} \min_{i\in\mathcal{I}} \min_{|\mathcal{B}|\leq k'} F_i(\mathcal{A} \setminus \mathcal{B}).$$

Another important problem in machine learning is feature selection. In feature selection, the goal is to select a subset of features which are informative with respect to, e.g., a given classification task. One objective frequently considered is the problem of selecting a set of features which maximize the information gained about the class variable $\mathcal{X}_Y$ after observing the features $\mathcal{X}_\mathcal{A}$, $F(\mathcal{A}) = H(\mathcal{X}_Y) - H(\mathcal{X}_Y \mid \mathcal{X}_\mathcal{A})$, where H denotes the Shannon entropy. Krause and Guestrin (2005) show, that in a large class of graphical models, the information

gain $F(\mathcal{A})$ is in fact a submodular function. Now we can consider a setting, where an adversary can delete features which we selected (as considered, e.g., by Globerson and Roweis 2006). The problem of selecting features robustly against such arbitrary deletion of, e.g., m features, is hence equivalent to the problem of maximizing $\min_{|\mathcal{B}|\leq m} F_{\mathcal{B}}(\mathcal{A})$, where $\mathcal{B}$ are the deleted features.

6.5.1 Improved Guarantees for Sensor Failures

As discussed above, in principle, we could find a placement robust to single sensor failures by using SATURATE to (approximately) solve

$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}|\leq k} \min_s F_s(\mathcal{A}).$$

However, since $|\mathcal{V}|$ can be very large, and the approximation guarantee α depends logarithmically on $|\mathcal{V}|$, such a direct approach might not be desirable. We can improve the guarantee from $\mathcal{O}(\log|\mathcal{V}|)$ to $\mathcal{O}(\log(k\log|\mathcal{V}|))$, which typically is much tighter, if $k \ll |\mathcal{V}|$ (i.e., we place far fewer sensors than we have possible sensor locations. We can improve the approximation guarantee drastically by noticing that $F_s(\mathcal{A}) = F(\mathcal{A})$ if $s \notin \mathcal{A}$. Hence,

$$\overline{F}_c(\mathcal{A}) = \frac{|\mathcal{V}| - |\mathcal{A}|}{|\mathcal{V}|} \min\{F(\mathcal{A}), c\} + \frac{1}{|\mathcal{V}|} \sum_{s\in\mathcal{A}} \widehat{F}_{s,c}(\mathcal{A}).$$

We can replace this objective by a new objective function,

$$\overline{F}'_c(\mathcal{A}) = \frac{k' - |\mathcal{A}|}{k'} \min\{F(\mathcal{A}), c\} + \frac{1}{k'} \sum_{s\in\mathcal{A}} \widehat{F}_{s,c}(\mathcal{A})$$

for some constant k' to be specified below. This modified objective is still monotonic and submodular when restricted to sets of size at most k'. It still holds that, for all subsets $|\mathcal{A}| \leq k'$, that

$$\overline{F}'_c(\mathcal{A}) \geq c \Leftrightarrow F_s(\mathcal{A}) \geq c \text{ for all } s \in \mathcal{V}.$$

How large should we choose k'? We have to choose k' large enough such that SATURATE will never choose sets larger than k'. A sufficient choice for k' is hence $\lceil \alpha k \rceil$, where $\alpha = 1 + \log\left(|\mathcal{V}| \max_{s\in\mathcal{V}} F(\{s\})\right)$. For this choice of k', our new approximation guarantee will be

$$\begin{aligned}\alpha' &= 1 + \log\left(\alpha k \max_{s\in\mathcal{V}} F(\{s\})\right) = 1 + \log\left(\left(1 + \log\left(|\mathcal{V}| \max_{s\in\mathcal{V}} F(\{s\})\right)\right) k \max_{s\in\mathcal{V}} F(\{s\})\right) \\ &\leq 1 + 2\log\left(k\log(|\mathcal{V}|) \max_{s\in\mathcal{V}} F(\{s\})\right)\end{aligned}$$

Hence, for the new objective $\overline{F}'_c$, we get a tighter approximation guarantee, $\alpha' = 1 + 2\log\left(k\log(|\mathcal{V}|)\max_{s\in\mathcal{V}} F(\{s\})\right)$, which now depends logarithmically on $k\log|\mathcal{V}|$, instead of the number of available locations $|\mathcal{V}|$. Note that this same approach can also provide tighter approximation guarantees in the case of multiple sensor failures.

7 Extensions

We now show how some of the assumptions made in our presentation above can be relaxed. We also discuss several extensions, allowing more complex cost functions, and the tradeoff between worst-case and average-case scores.

7.1 Non-integral Objectives

In our analysis of SATURATE (Section 4), we have assumed, that each of the objective functions F_i only take values in the positive integers. However, most objective functions of interest in observation selection (such as those discussed in Section 6) typically do not meet this assumption. If the F_i take on rational numbers, we can scale the objectives by multiplying by their common denominator.

If we allow small additive approximation error (i.e., are indifferent if the approximate solution differs from the optimal solution in low order bits), we can also approximate the values assumed by the functions F_i by their highest order bits. In this case, we replace the functions $F_i(\mathcal{A})$ by the approximations

$$F'_i(\mathcal{A}) = \frac{\lceil 2^j F_i(\mathcal{A}) \rceil}{2^j}.$$

By construction, $F'_i(\mathcal{A}) \leq F_i(\mathcal{A}) \leq F'_i(\mathcal{A})(1 + 2^{-j})$, i.e., F'_i is within a factor of $(1 + 2^{-j})$ of F_i. Also, $2^j F'_i(\mathcal{A})$ is integral. However, $F'_i(\mathcal{A})$ is not guaranteed to be submodular. Nevertheless, an analysis similar to the one presented by Krause et al. (2007b) can be used to bound the effect of this approximation on the theoretical guarantees α obtained by the algorithm, which will now scale linearly with the number j of high order bits considered. In practice, as we show in Section 8, SATURATE provides state-of-the-art performance, even without rounding the objectives to the highest order bits.

7.2 Non-constant Thresholds

Consider the example of minimizing the maximum variance in Gaussian Process regression. Here, the $F_i(\mathcal{A}) = \sigma_i^2 - \sigma_{i|\mathcal{A}}^2$ denote the variance reductions at location i. However, rather than guaranteeing that $F_i(\mathcal{A}) \geq c$ for all i (which, in this example, means that the *minimum variance reduction* is c), we want to guarantee that $\sigma_{i|\mathcal{A}}^2 \leq c$ for all i. We can easily adapt our approach to handle this case: Instead of defining $\widehat{F}_{i,c}(\mathcal{A}) = \min\{F_i(\mathcal{A}), c\}$, we define $\widehat{F}_{i,c}(\mathcal{A}) = \min\{F_i(\mathcal{A}), \sigma_i^2 - c\}$, and then again perform binary search over c, but searching for the smallest c instead. The algorithm, using objectives modified in this way, will bear the same approximation guarantees.

7.3 Non-uniform Observation Costs

We can extend SATURATE to the setting where different observations have different costs. In the spatial monitoring setting for example, certain locations might be more expensive to acquire a measurement from. Suppose a cost function $g : \mathcal{V} \to \mathbb{R}^+$ assigns each element $s \in \mathcal{V}$ a positive cost $g(s)$; the cost of a set of observations is then $g(\mathcal{A}) = \sum_{s \in \mathcal{A}} g(s)$. The

problem is to find $\mathcal{A}^* = \text{argmax}_{\mathcal{A} \subset \mathcal{V}} \min_i F_i(\mathcal{A})$ subject to $g(\mathcal{A}) \leq B$, where $B > 0$ is a *budget* we can spend on making observations. In this case, we use the rule

$$\delta_s \leftarrow \frac{\overline{F}_c(\mathcal{A} \cup \{s\}) - \overline{F}_c(\mathcal{A})}{g(s)}$$

in Algorithm 1. For this modified algorithm, Theorem 5 still holds, with $|\mathcal{A}|$ replaced by $g(\mathcal{A})$ and k replaced by B. This more general result holds, since the analysis of the greedy algorithm for submodular covering of Wolsey (1982), which we used to prove Lemma 4, applies to the more general setting of non-uniform cost functions.

7.4 Handling More Complex Cost Functions

So far, we considered problems where we are given an *additive* cost function $g(\mathcal{A})$ over the possible sets $\mathcal{A}$ of observations. In some applications, more complex cost functions arise. For example, when placing wireless sensor networks, the placements $\mathcal{A}$ should not only be informative (i.e., $F_i(\mathcal{A})$ should be high for all utility functions F_i), but the placement should also have *low communication cost*. Krause et al. (2006) describe such an approach, where the cost $g(\mathcal{A})$ measures the *expected number of retransmissions* required for sending messages across an optimal routing tree connecting the sensors $\mathcal{A}$. Formally, the observations s are considered to be nodes in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with edge weights $w(e)$ for each edge $e \in \mathcal{E}$. The cost $g(\mathcal{A})$ is the cost of a minimum Steiner Tree (*cf.,* Vazirani 2003) connecting the observations $\mathcal{A}$ in the graph $\mathcal{G}$.

More generally, we want to solve problems of the form

$$\operatorname*{argmax}_{\mathcal{A}} \min_i F_i(\mathcal{A}) \text{ subject to } g(\mathcal{A}) \leq B, \tag{7.1}$$

where $g(\mathcal{A})$ is a complex cost function. The key insight of the SATURATE algorithm is that the non-submodular robust optimization problem can be approximately solved by solving a submodular covering problem. In the case where $g(\mathcal{A}) = |\mathcal{A}|$ this problem requires solving (4.2). More generally, we can apply SATURATE to any problem where we can (approximately) solve

$$\mathcal{A}_c = \operatorname*{argmin}_{\mathcal{A} \subseteq \mathcal{V}} g(\mathcal{A}), \quad \text{such that} \quad \overline{F}_c(\mathcal{A}) = c. \tag{7.2}$$

Problem (7.2) can be (approximately) solved for a variety of cost functions, such as those arising from communication constraints (Krause et al., 2006) and path constraints (Singh et al., 2007; Meliou et al., 2007).

Let us summarize our analysis as follows:

Proposition 7 *Assume we have an algorithm which, given a monotonic submodular function F and a cost function g, returns a solution $\mathcal{A}'$ such that $F(\mathcal{A}') = F(\mathcal{V})$ and*

$$g(\mathcal{A}') \leq \alpha_F \min_{\mathcal{A}: F(\mathcal{A}) = F(\mathcal{V})} g(\mathcal{A}),$$

where α_F depends on the function F. SATURATE*, using this covering algorithm, can obtain a solution $\mathcal{A}_S$ to the* RSOS *problem such that*

$$\min_i F_i(\mathcal{A}_S) \geq \max_{g(\mathcal{A}) \leq B} \min_i F_i(\mathcal{A}),$$

and

$$g(\mathcal{A}_S) \leq \alpha_{\overline{F}} B,$$

where $\alpha_{\overline{F}}$ *is the approximation factor of the covering algorithm, when applied to* $\overline{F} = \frac{1}{m}\sum_i F_i$.

Note that the formalism developed in this section also allows to handle robust versions of combinatorial optimization problems such as the *Knapsack* (*cf.,* Martello and Toth, 1990), Orienteering (*cf.,* Laporte and Martello, 1990; Blum et al., 2003) and *Budgeted Steiner Tree* (*cf.,* Johnson et al., 2000) problems. In these problems, instead of a general *submodular* objective function, the special case of a *modular* (additive) function F is optimized:

$$\mathcal{A}^* = \operatorname*{argmax}_{g(\mathcal{A}) \leq B} F(\mathcal{A}).$$

The problems differ only in the choice of the complex cost function. In *Knapsack* for example, g is additive, in the *Budgeted Steiner Tree* problem, $g(\mathcal{A})$ is the cost of a minimum Steiner tree connecting the nodes $\mathcal{A}$ in a graph, and in *Orienteering*, $g(\mathcal{A})$ is the cost of a shortest path connecting the nodes $\mathcal{A}$ in a graph. In practice, often the utility function $F(\mathcal{A})$ is not exactly known, and a solution is desired which is robust against worst-case choice of the utility function. Since modular functions are a special case of submodular functions, such problems can be approximately solved using Proposition 7.

7.5 Trading Off Average-case and Worst-case Scores

In some applications, optimizing the worst-case score $F_{wc}(\mathcal{A}) = \min_i F_i(\mathcal{A})$ might be a too pessimistic approach. On the other hand, ignoring the worst-case and only optimizing the average-case (the expected score under a distribution over the objectives) $F_{ac}(\mathcal{A}) = \frac{1}{m}\sum_i F_i(\mathcal{A})$ might be too optimistic. In fact, in Section 8 we show that optimizing the average-case score F_{ac} can often lead to drastically poor worst-case scores. In general, we might be interested in solutions, which perform well both in the average- and worst-case scores.

Formally, we can define a multicriterion optimization problem, where we intend to optimize the *vector* $[F_{ac}(\mathcal{A}), F_{wc}(\mathcal{A})]$. In this setting, we can only hope for *Pareto-optimal* solutions (*cf.,* Boyd and Vandenberghe, 2004, in the context of convex functions). A set $\mathcal{A}^*$, $|\mathcal{A}^*| \leq k$ is called *Pareto-optimal*, if it is not *dominated*, i.e., there does not exist another set $\mathcal{B}$, $|\mathcal{B}| \leq k$ with $F_{ac}(\mathcal{B}) > F_{ac}(\mathcal{A}^*)$ and $F_{wc}(\mathcal{B}) \geq F_{wc}(\mathcal{A}^*)$ (or $F_{ac}(\mathcal{B}) \geq F_{ac}(\mathcal{A}^*)$ and $F_{wc}(\mathcal{B}) > F_{wc}(\mathcal{A}^*)$).

One possible approach to find such Pareto-optimal solutions is constrained optimization[2]: for a specified value of c_{ac}, we desire a solution to

$$\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}| \leq k} F_{wc}(\mathcal{A}) \text{ such that } F_{ac}(\mathcal{A}) \geq c_{ac}. \tag{7.3}$$

By specifying different values of c_{ac} in (7.3), we would obtain different Pareto-optimal solutions[3]. Figure 5 presents an example of several Pareto-optimal solutions, based on data

[2]Another approach is *scalarization*, where we optimize $F_\lambda(\mathcal{A}) = \lambda F_{wc}(\mathcal{A}) + (1-\lambda)F_{ac}(\mathcal{A})$ for some λ, $0 < \lambda < 1$. SATURATE can be modified to handle such scalarized objectives as well (*cf.,* Appendix B).

[3]In fact, *all* Pareto-optimal solutions can be found in this way (Papadimitriou and Yannakakis, 2000).

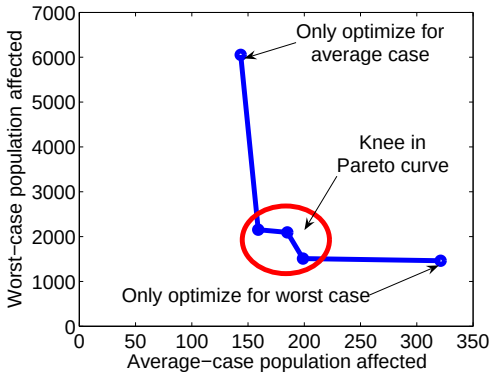


Figure 5: Tradeoff curve for simultaneously optimizing the average- and worst-case score in the water distribution network monitoring application. Notice the knee in the tradeoff curve, indicating that by performing multi-criterion optimization, solutions performing well for both average- and worst-case scores can be obtained.

from the outbreak detection problem (Details will be discussed in Section 8.3). This curve shows that, using the techniques described below, multicriterion solutions can be found which combine the advantages of worst-case and average-case solutions.

We can modify SATURATE to solve Problem (7.3) in the following way. Let us again assume we know the optimal value c_{wc} achievable for Problem (7.3). Then, Problem (7.3) is equivalent to solving

$$\mathcal{A}^* = \operatorname*{argmin}_{\mathcal{A}} |\mathcal{A}| \text{ subject to } F_{wc}(\mathcal{A}) \geq c_{wc} \text{ and } F_{ac}(\mathcal{A}) \geq c_{ac}. \tag{7.4}$$

Now, using our notation from Section 4, this problem is again equivalent to

$$\mathcal{A}^* = \operatorname*{argmin}_{\mathcal{A}} |\mathcal{A}| \text{ subject to } \overline{F}_{c_{wc},c_{ac}} = c_{wc} + c_{ac}, \tag{7.5}$$

where

$$\overline{F}_{c_{wc},c_{ac}}(\mathcal{A}) = \overline{F}_{c_{wc}}(\mathcal{A}) + \min\{F_{ac}(\mathcal{A}), c_{ac}\}.$$

Note that $\overline{F}_{c_{wc},c_{ac}}$ is a submodular function, and hence (7.5) is a submodular covering problem, which can be approximately solved using the greedy algorithm.

For any choice of c_{ac}, we can find the optimal value of c_{wc} by performing binary search on c_{wc}. We summarize our analysis in the following Theorem:

Theorem 8 *For any integer k and constraint c_{ac},* SATURATE *finds a solution $\mathcal{A}_S$ (if it exists) such that*

$$F_{wc}(\mathcal{A}_S) \geq \max_{|\mathcal{A}| \leq k, F_{ac}(\mathcal{A}) \geq c_{ac}} F_{wc}(\mathcal{A}),$$

$F_{ac}(\mathcal{A}_S) \geq c_{ac}$, and $|\mathcal{A}_S| \leq \alpha k$, for $\alpha = 1 + \log\left(2 \max_{s \in \mathcal{V}} \sum_i F_i(s)\right)$. Each such solution $\mathcal{A}_S$ is approximately *Pareto-optimal, i.e., there does not exist a set $\mathcal{B}$, $|\mathcal{B}| \leq k$ such that $\mathcal{B}$ dominates $\mathcal{A}_S$. The total number of submodular function evaluations is $\mathcal{O}\left(|\mathcal{V}|^2 m \log(\sum_i F_i(\mathcal{V}))\right)$.*

8 Experimental Results

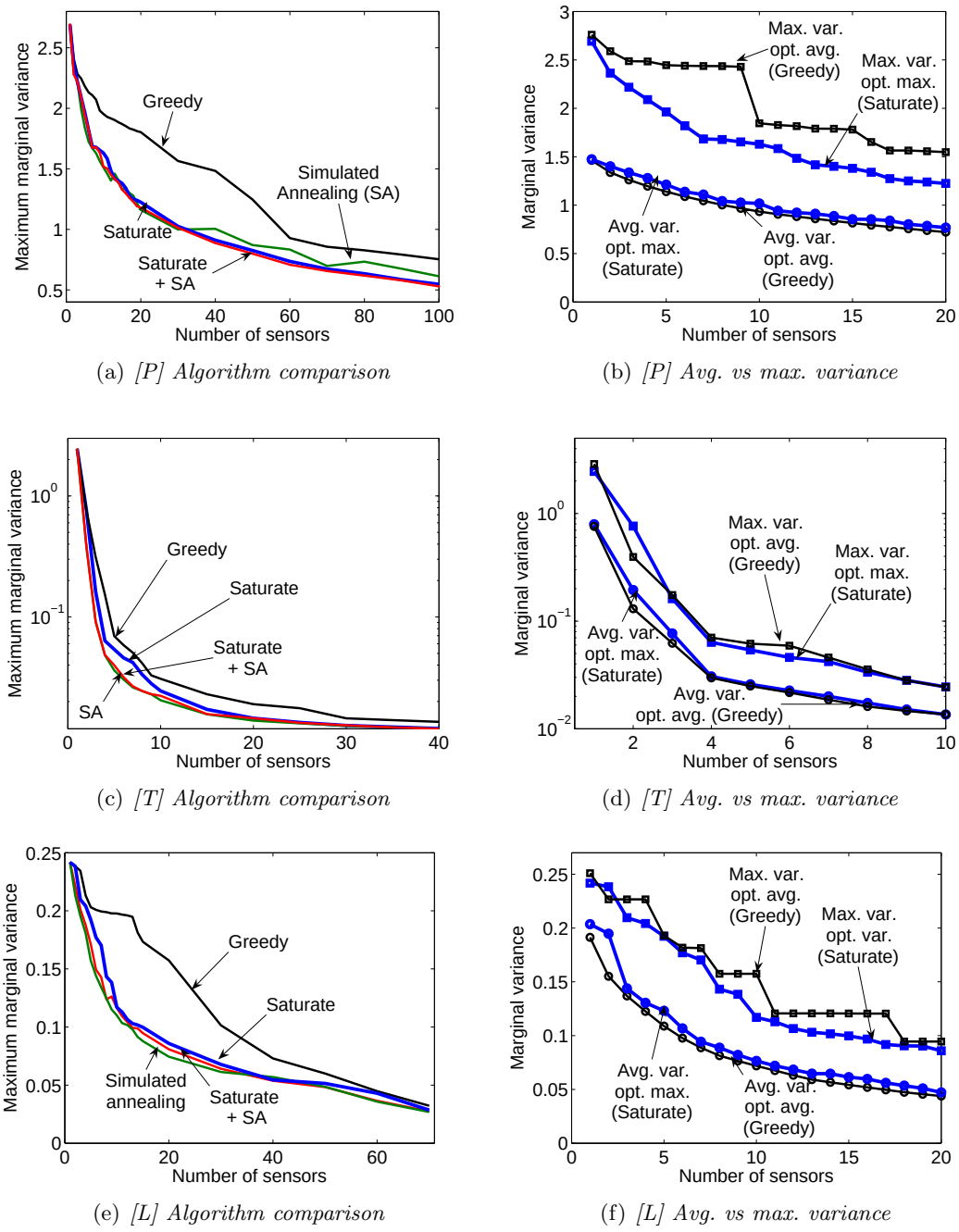
8.1 Minimizing the Maximum Kriging Variance

First, we use SATURATE to select observations in a GP to minimize the maximum posterior variance (*cf.,* Section 6.1). We consider three data sets: [T] temperature data from a deployment of 52 sensors at Intel Research Berkeley, [P] Precipitation data from the Pacific Northwest of the United States (Widmann and Bretherton, 1999) and [L] temperature data from the NIMS sensor node (Harmon et al., 2006) deployed at a lake near the University of California, Merced. For the three monitoring problems, [T], [P], and [L], we discretize the space into 46, 167 and 86 locations each, respectively. For [T], we consider the empirical covariance matrix of temperature sensor measurements obtained over a period of 5 days. For [P], we consider the empirical covariance of 50 years of data, which we preprocessed as described by Krause et al. (2007b). For [L], we train a nonstationary Gaussian Process using data from a single scan of the lake by the NIMS sensor node, using a method described by Krause and Guestrin (2007b).

In the geostatistics literature, the predominant choice of optimization algorithms for selecting locations in a GP to minimize the (maximum and average) predictive variance are carefully tuned local search procedures, prominently simulated annealing (*cf.,* Sacks and Schiller 1988; Wiens 2005; van Groenigen and Stein 1998). We compare our SATURATE algorithm against a state-of-the-art implementation of such a simulated annealing (SA) algorithm, first proposed by Sacks and Schiller (1988). We use an optimized implementation described recently by Wiens (2005). This algorithm has 7 parameters which need to be tuned, describing the annealing schedule, distribution of iterations among several inner loops, etc. We use the parameter settings as reported by Wiens (2005), and present the best result of the algorithm among 10 random trials. In order to compare observation sets of the same size, we called SATURATE with $\alpha = 1$.

Figures 6(a), 6(c) and 6(e) compare simulated annealing, SATURATE, and the greedy algorithm which greedily selects elements which decrease the maximum variance the most on the three data sets. We also used SATURATE to initialize the simulated annealing algorithm (using only a single run of simulated annealing, as opposed to 10 random trials). In all three data sets, SATURATE obtains placements which are drastically better than the placements obtained by the greedy algorithm. Furthermore, the performance is very close to the performance of the simulated annealing algorithm. In our largest monitoring dataset [P], SATURATE even strictly outperforms the simulated annealing algorithm when selecting 30 and more sensors. Furthermore, as Figure 7 shows, SATURATE is significantly faster than simulated annealing, by factors of 5-10 for larger problems. When using SATURATE in order to initialize the simulated annealing algorithm, the resulting performance almost always resulted in the best solutions we were able to find with any method, while still executing faster than simulated annealing with 10 random restarts as proposed by Wiens (2005). These results indicate that SATURATE compares favorably to state-of-the-art local search heuristics, while being faster, requiring no parameters to tune, and providing theoretical approximation guarantees.

Optimizing for the maximum variance could potentially be considered too pessimistic. Hence we compared placements obtained by SATURATE, minimizing the maximum marginal posterior variance, with placements obtained by the greedy algorithm, where we minimize



(a) *[P] Algorithm comparison*

(b) *[P] Avg. vs max. variance*

(c) *[T] Algorithm comparison*

(d) *[T] Avg. vs max. variance*

(e) *[L] Algorithm comparison*

(f) *[L] Avg. vs max. variance*

Figure 6: (a,c,e) SATURATE, greedy and SA on the (a) precipitation, (b) building temperature and (c) lake temperature data. SATURATE performs comparably with the fine-tuned SA algorithm, and outperforms it for larger placements. (b,d,f) Optimizing for the maximum variance (using SATURATE) leads to low average variance, but optimizing for average variance (using greedy) does not lead to low maximum variance.

the *average* marginal variance. Note, that, whereas the maximum variance reduction is non-submodular, the *average* variance reduction is (often) submodular (Das and Kempe,

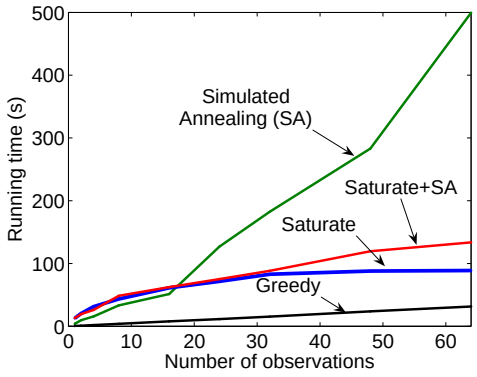


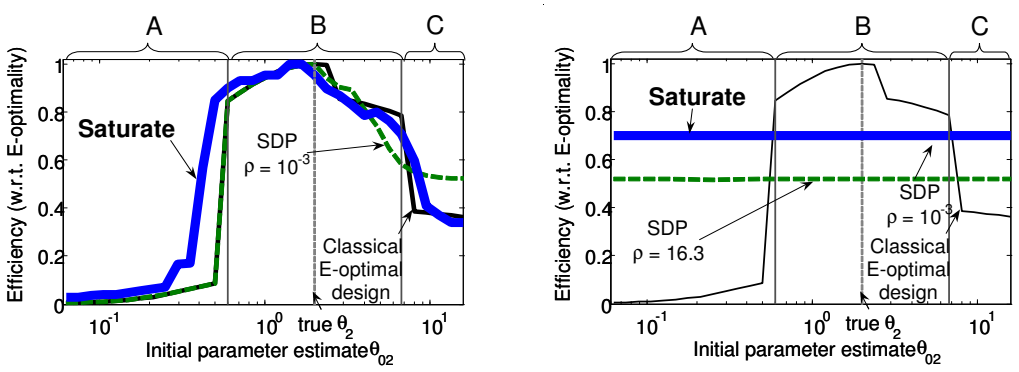
Figure 7: Running time for algorithms on the precipitation data set [P].

2007), and hence the greedy algorithm can be expected to provide near-optimal placements. Figures 6(b), 6(d) and 6(f) present the maximum and average marginal variances for both algorithms. On all three data sets, our results show that if we optimize for the maximum variance we still achieve comparable average variance. If we optimize for average variance however, the maximum posterior variance remains much higher.

8.2 Robust Experimental Design

We consider the robust design of experiments (*cf.,* Section 6.3) for the Michaelis-Menten mass-action kinetics model, as discussed by Flaherty et al. (2006). The goal is least-square parameter estimation for a function $y = f(x, \theta)$, where x is the chosen experimental stimulus (the initial substrate concentration S_0), and $\theta = (\theta_1, \theta_2)$ are two parameters as described by Flaherty et al. (2006). The stimulus x is chosen from a menu of six options, $x \in \{1/8, 1, 2, 4, 8, 16\}$, each of which can be repeatedly chosen. The goal is to produce a fractional design $\mathbf{w} = (w_1, \ldots, w_6)$, where each component w_i measures the relative frequency according to which the stimulus x_i is chosen. Since f is nonlinear, f is linearized around an initial parameter estimate $\theta_0 = (\theta_{01}, \theta_{02})$, and approximated by its Jacobian V_{θ_0}. Classical experimental design considers the error covariance of the least squares estimate $\hat{\theta}$, $\mathrm{Cov}(\hat{\theta} \mid \theta_0, \mathbf{w}) = \sigma^2 (V_{\theta_0}^T W V_{\theta_0})^{-1}$, where $W = \mathrm{diag}(\mathbf{w})$, and aims to find designs $\mathbf{w}$ which minimize this error covariance. E-optimality, the criterion adopted by Flaherty et al. (2006), measures smallness in terms of the maximum eigenvalue of the error covariance matrix. The optimal $\mathbf{w}$ can be found using Semidefinite Programming (SDP) (Boyd and Vandenberghe, 2004).

The estimate $\mathrm{Cov}(\hat{\theta} \mid \theta_0, \mathbf{w})$ depends on the initial parameter estimate θ_0, where linearization is performed. However, since the goal is parameter estimation, a "certain circularity is involved" (Flaherty et al., 2006). To avoid this problem, Flaherty et al. (2006) find a design $\mathbf{w}_\rho(\theta_0)$ by solving a robust SDP which minimizes the error size, subject to a worst-case perturbation Δ on the Jacobian V_{θ_0}; the robustness parameter ρ bounds the spectral norm of Δ. As evaluation criterion, Flaherty et al. (2006) define a notion of *efficiency*, which is the error size of the optimal design with correct initial parameter estimate, divided by the error when using a robust design obtained at the wrong initial parameter estimates, i.e.,



(a) *Low uncertainty in* θ_0 (b) *High uncertainty in* θ_0

Figure 8: Efficiency of robust SDP of Flaherty et al. (2006) and SATURATE on a biological experimental design problem. (a) Low assumed uncertainty in initial parameter estimates: SDP performs better in region C, SATURATE performs better in region A. (b) High assumed uncertainty in initial parameter estimates: SATURATE outperforms the SDP solutions.

$$\text{efficiency} \equiv \frac{\lambda_{\max}[\text{Cov}(\hat{\theta} \mid \theta_{true}, \mathbf{w}_{opt}(\theta_{true}))]}{\lambda_{\max}[\text{Cov}(\hat{\theta} \mid \theta_{true}, \mathbf{w}_{\rho}(\theta_0))]},$$

where $\mathbf{w}_{opt}(\theta)$ is the E-optimal design for parameter θ. They show that for appropriately chosen values of ρ, the robust design is more *efficient* than the optimal design, if the initial parameter θ_0 does not equal the true parameter.

While their results are very promising, an arguably more natural approach than perturbing the Jacobian would be to perturb the initial parameter estimate, around which linearization is performed. For example, if the function f describes a process which behaves characteristically differently in different "phases", and the parameter θ controls which of the phases the process is in, then a robust design should intuitively "hedge" the design against the behavior in each possible phase. In such a case, the uniform distribution (which the robust SDP chooses for large ρ) would not be the most robust design.

If we discretize the space of possible parameter perturbations (within a reasonably chosen interval), we can use SATURATE to find robust experimental designs. While the classical E-optimality is not submodular (Krause et al., 2007b), Bayesian A-optimality is (usually) submodular (Das and Kempe, 2007; Krause et al., 2007b). Here, the goal is to minimize the *trace* instead of maximum eigenvalue size of the covariance matrix. Furthermore, we equip the parameters θ with an uninformative normal prior (which we chose as $\text{diag}([20^2, 20^2])$) as typically done in Bayesian experimental design. We then minimize the expected trace of the posterior error covariance, $\text{tr}(\Sigma_{\theta|\mathcal{A}})$. Hereby, $\mathcal{A}$ is a discrete design of 20 experiments, where each option x_i can be chosen repeatedly. In order to apply SATURATE, for each θ_0, we define $F_{\theta_0}(\mathcal{A})$ as the normalized variance reduction

$$F_{\theta_0}(\mathcal{A}) = \frac{1}{Z_{\theta_0}} \left(\text{tr}\left(\Sigma_{\theta}^{(\theta_0)}\right) - \text{tr}\left(\Sigma_{\theta|\mathcal{A}}^{(\theta_0)}\right) \right).$$

The normalization Z_{θ_0} is chosen such that $F_{\theta_0}(\mathcal{A}) = 1$ if

$$\mathcal{A} = \operatorname*{argmax}_{|\mathcal{A}'|=20} F_{\theta_0}(\mathcal{A}'),$$

i.e., if $\mathcal{A}$ is chosen to maximize only F_{θ_0}. SATURATE is then used to maximize the worst-case normalized variance reduction.

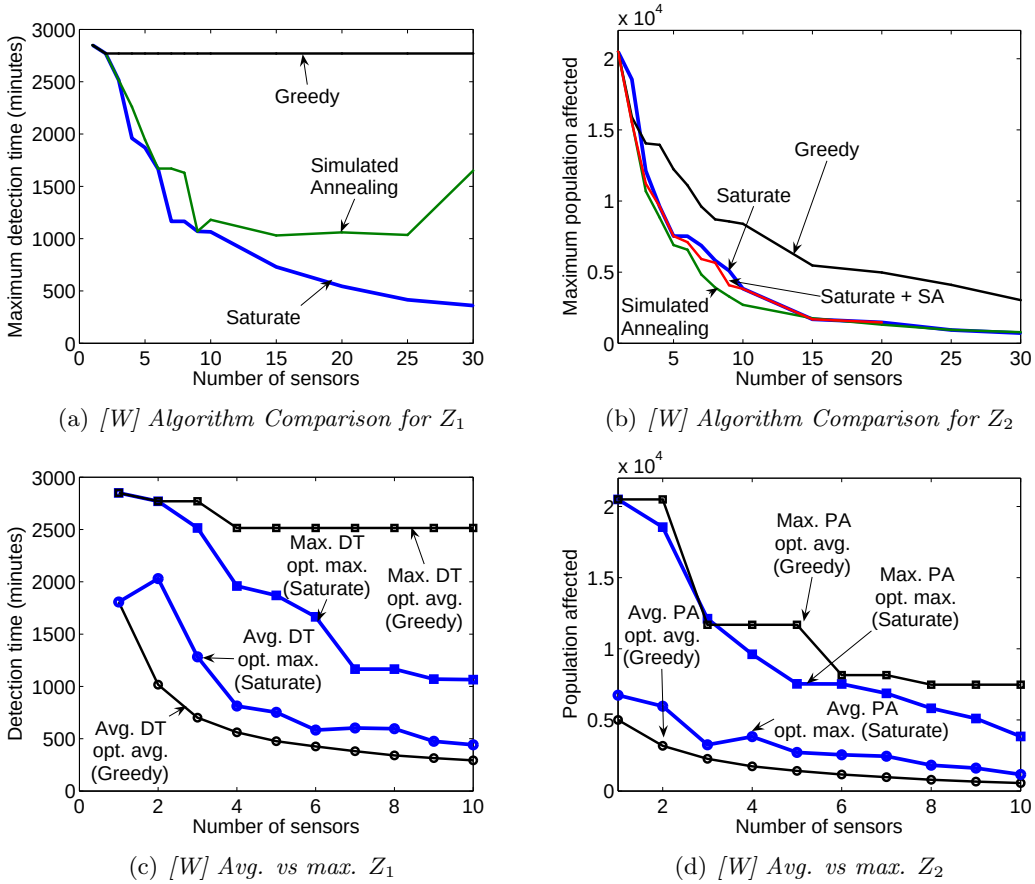
We reproduced the experiment of Flaherty et al. (2006), where the initial estimate of the second component θ_{02} of θ_0 was varied between 0 and 16, the "true" value being $\theta_2 = 2$. For each initial estimate of θ_{02}, we computed a robust design, using the SDP approach and using SATURATE, and compared them using the efficiency metric of Flaherty et al. (2006). Note that this efficiency metric is defined with respect to E-optimality, even though we optimize Bayesian A-optimality, hence potentially putting SATURATE at a disadvantage. We first optimized designs which are robust against a small perturbation of the initial parameter estimate. For the SDP, we chose a robustness parameter $\rho = 10^{-3}$, as reported in Flaherty et al. (2006). For SATURATE, we considered an interval around $[\theta\frac{1}{1+\varepsilon}, \theta(1+\varepsilon)]$, discretized in a 5×5 grid, with $\varepsilon = .1$.

Figure 8(a) shows three characteristically different regions, A, B, C, separated by vertical lines. In region B which contains the true parameter setting, the E-optimal design (which is optimal if the true parameter is known, i.e., $\theta_{02} = \theta_2$) performs similar to both robust methods. Hence, in region B (i.e., small deviation from the true parameter), robustness is not really necessary. Outside of region B however, where the standard E-optimal design performs badly, both robust designs do not perform well either. This is an intuitive result, as they were optimized to be robust only to small parameter perturbations.

Consequently, we compared designs which are robust against a *large* parameter range. For SDP, we chose $\rho = 16.3$, which is the maximum spectral variation of the Jacobian when we consider all initial estimates from θ_{02} varying between 0 and 16. For SATURATE, we optimized a single design which achieves the maximum normalized variance reduction over all values of θ_{02} between 0 and 16. Figure 8(b) shows, that in this case, the design obtained by SATURATE achieves an efficiency of 69%, whereas the efficiency of the SDP design is only 52%. In the regions A and C, the SATURATE design strictly outperforms the other robust designs. This experiment indicates that designs which are robust against a large range of initial parameter estimates, as provided by SATURATE, can be more efficient than designs which are robust against perturbations of the Jacobian (the SDP approach).

8.3 Outbreak Detection

Consider a city water distribution network, delivering water to households via a system of pipes, pumps, and junctions. Accidental or malicious intrusions can cause contaminants to spread over the network, and we want to select a few locations (pipe junctions) to install sensors, in order to detect these contaminations as quickly as possible (*cf.,* Section 6.3). In August 2006, the Battle of Water Sensor Networks (BWSN) (Ostfeld et al., 2006) was organized as an international challenge to find the best sensor placements for a real (but anonymized) metropolitan water distribution network, consisting of 12,527 nodes. In this challenge, a set of intrusion scenarios is specified, and for each scenario a realistic simulator provided by the EPA (Rossman, 1999) is used to simulate the spread of the contaminant for a 48 hour period. An intrusion is considered detected when one selected node shows positive contaminant concentration. BWSN considered a variety of impact measures, including the time to detection (called Z_1), and the size of the affected population calculated using



(a) *[W] Algorithm Comparison for* Z_1

(b) *[W] Algorithm Comparison for* Z_2

(c) *[W] Avg. vs max.* Z_1

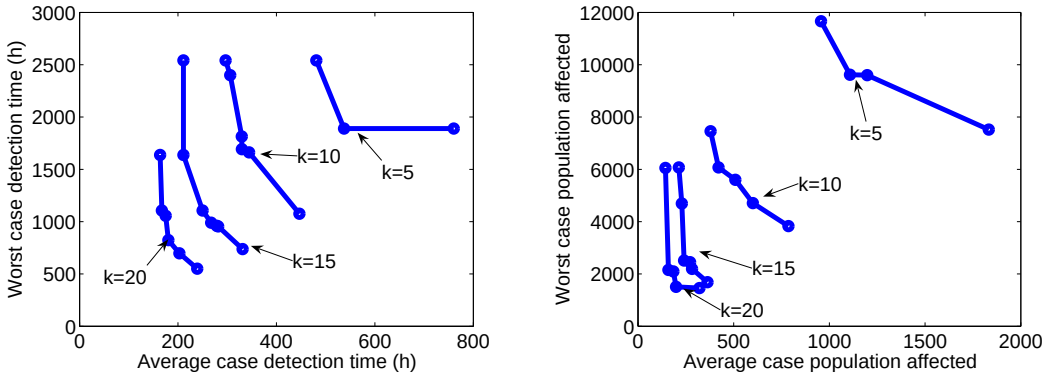
(d) *[W] Avg. vs max.* Z_2

Figure 9: (a,b) compare SATURATE, greedy and SA in the water network setting, when optimizing worst-case detection time (Z_1, (a)) and affected population (Z_2, (b)). SATURATE performs comparably to SA for Z_2 and strictly outperforms SA for Z_1. (c,d) compare optimizing for the worst-case vs. average-case objectives. Optimizing for the worst-case leads to good average case performs, but not vice versa.

a realistic disease model (Z_2). The goal of BWSN was to minimize the *expectation* of the impact measures Z_1 and Z_2 given a *uniform distribution* over intrusion scenarios.

In this paper, we consider the *adversarial* setting, where an opponent chooses the contamination scenario with knowledge of the sensor locations. The objective functions Z_1 and Z_2 are in fact submodular for a fixed intrusion scenario (Leskovec et al., 2007), and so the robust optimization problem of minimizing the impact of the worst possible intrusion fits into our formalism. For these experiments, we consider scenarios which affect at least 10% of the network, resulting in a total of 3424 scenarios. Figures 9(a) and 9(b) compare the greedy algorithm, SATURATE and the simulated annealing (SA) algorithm for the problem of maximizing the worst-case detection time (Z_1) and worst-case affected population (Z_2).

Interestingly, the behavior is very different for the two objectives. For the affected population (Z_2), greedy performs reasonably, and SA sometimes even outperforms SATURATE. For the detection time (Z_1), however, the greedy algorithm did not improve the objective at all, and SA performs poorly. The reason is that for Z_2, the maximum achievable scores, $F_i(\mathcal{V})$, vary drastically, since some scenarios have much higher impact than others. Hence,



(a) *[W] Pareto curve for Z_1*

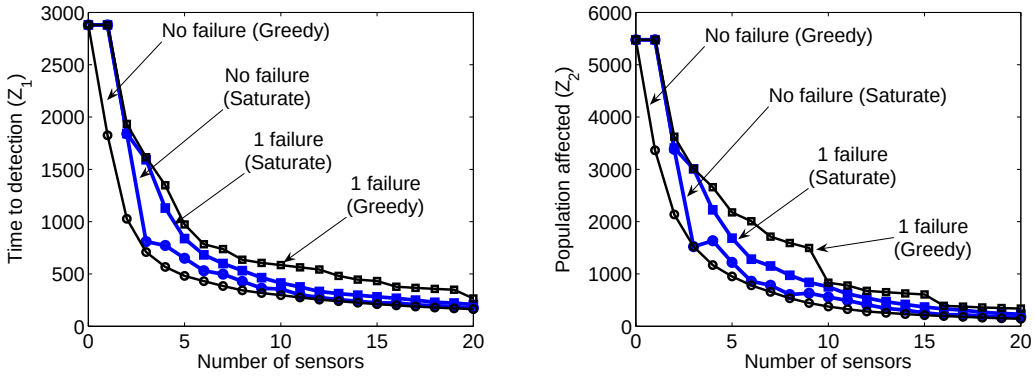
(b) *[W] Pareto curve for Z_2*

Figure 10: Experiments on trading off worst-case and average-case penalties on the water network [W] data, minimizing detection time (a) and affected population (b).

there is a strong "gradient", as the worst-case objective changes quickly when the high impact scenarios are covered. This gradient allows greedy and SA to work well. On the contrary, for Z_1, the maximum achievable scores, $F_i(\mathcal{V})$, are constant, since all scenarios have the same simulation duration. Unless *all* scenarios are detected, the worst-case detection time stays constant at the simulation length. Hence, many node exchange proposals considered by SA, as well as the addition of a new sensor location by greedy, do not change the worst-case objective, and the algorithms have no useful performance metric.

Figures 9(c) and 9(d) compare the placements of SATURATE (when optimizing the worst-case penalty), and greedy (when optimizing the average-case penalty, which is submodular). Similarly to the results in the GP setting, optimizing the worst-case score leads to reasonable performance in the average case score, but not necessarily vice versa (especially when considering the detection time).

We also performed experiments trading off the worst-case and average-case penalty reductions, using the approach discussed in Section 7.5. We first ran the greedy algorithm to optimize the average-case score, and then ran SATURATE to optimize the worst-case score. We considered the average-case scores c_{ac}^{greedy} and $c_{ac}^{Saturate}$ obtained by both algorithms, and uniformly discretized the interval bounded by these average-case scores. For each score level c_{ac} in the discretization, we use the modified SATURATE algorithm as described in Section 7.5, maximizing the worst-case score, subject to a constraint on the average-case score. Each possible value of the constraint on c_{ac} can lead to a different solution, trading off average- and worst-case scores. Figure 10(a) presents the tradeoff curve obtained in this fashion for the detection time (Z_1) metric, for different numbers k of placed sensors. We generally observe that there is more variability in the worst-case score than in the average-case score. We can also see that when placing 5 sensors, there is a prominent knee in the tradeoff curve, effectively achieving the minimum worst-case penalty but drastically reducing the average-case penalty incurred when compared to only optimizing for the worst-case score. The other tradeoff curves do not exhibit quite such prominent knees, but nevertheless allow flexibility in trading off worst- and average-case scores. Figure 10(b) presents the same experiment for the population affected (Z_2) metric. Here, we notice prominent knees



(a) *[W] Sensor failures for* Z_1 (b) *[W] Sensor failures for* Z_2

Figure 11: (a,b) compare Greedy (ignoring sensor failures) and SATURATE (optimizing for the worst-case sensor failure) on water network data with detection time (a) and population affected (b) scores.

when placing $k = 15$ and 20 sensors. We can generally conclude that trading off average- and worst-case scores allows to effectively achieve a compromise between too pessimistic (only optimizing for the worst case) and optimistic (only optimizing for the average case) objectives.

8.4 Sensor Failures

We also performed experiments on analyzing worst-case sensor failures (*cf.,* Section 6.5). We consider the outbreak detection application, and optimize the average score, i.e., $F(\mathcal{A}) = \frac{1}{m}\sum_i F_i(\mathcal{A})$ (modeling, e.g., accidental contaminations). We use SATURATE in order to optimize the modified objective function $\overline{F}'_c$ described in Section 6.5.1, for increasing numbers of sensors k. We also use the greedy algorithm to optimize sensor placements, ignoring possible sensor failures. For both algorithms, we compute the expected scores (penalty reductions Z_1 and Z_2) in the case of no sensor failure, and in the case of a single, worst-case sensor failure. Figure 11(a) presents the results for the time to detection objective (Z_1). We can see, that initially, with small numbers of sensors, failures can strongly diminish the Z_1 score. However, as the number of sensors increases, the placement scores optimized using SATURATE for sensor failures quickly approach those of Greedy in the case of no sensor failures. Hence, even if only a small number of sensors are placed, SATURATE can quickly exploit redundancy and find sensor placements, which perform well both with and without sensor failures. On the other hand, when not taking sensor failures into account, such failures can drastically diminish the utility of a placed set of sensors. Figure 11(b) presents analogous results when minimizing the affected population (Z_2).

8.5 Parameter Uncertainty

We also conducted experiments on selecting variables under parameter uncertainty (*cf.,* Section 6.2). More specifically, we consider a sensor placement problem for monitoring temperature in a building. In such a problem, we would like to place sensors in order to

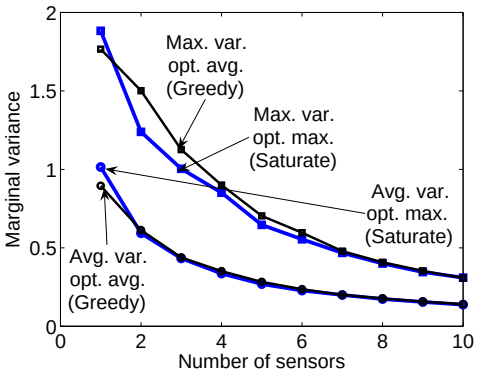


Figure 12: [T] Average and Maximum variance when optimizing for four different covariance models obtained during different parts of the day.

get accurate predictions at various times of the day. However, since phenomena such as temperature in buildings change over time, at different times of the day, different placements would be most informative.

In our experiment, we consider the temperature data set [T], and learn four models, described by parameters $\theta_1, \ldots, \theta_4$, during four six-hour time periods over the day: 12am-6am, 6am-12pm, 12pm-6pm and 6pm-12am. As models, we use the empirical covariances $\Sigma^{(\theta_i)}$ from the corresponding time periods of the 5 day historical training data. We also use the single model Σ for the entire day, as described in Section 8.1. We then use the greedy algorithm to optimize sensor placement of increasing sizes for the single model Σ, optimizing the average variance reduction objective function. Similarly, we use SATURATE to optimize the minimum variance reduction over the four models $\Sigma^{(i)}$, normalized by the average variance over the entire space.

Subsequently, we used both placements to compute the average Root Mean Squared (RMS) prediction error over the entire day on 2 days of held out test data. We also computed the maximum RMS error over the four six-hour time periods. Figure 12 presents the results of this experiment. While the average RMS error is roughly equal for both placements, the maximum RMS error is larger for the greedy sensor placement, as compared to the robust placement of SATURATE, especially for small numbers of sensors (six and less sensors).

9 Reducing the Number of Objective Functions

In many of the examples considered in Section 6, the number m of objective functions F_i can be quite large (e.g., one F_i per parameter setting, or outbreak scenario), which impacts both the running time (which depends linearly on m) and the approximation guarantees (which depend logarithmically on m) of SATURATE. Hence, showing that we can work with a smaller set of objectives has both computational and theoretical advantages.

9.1 Removal of Dominated Strategies

One direct approach to eliminate objective functions (and hence speed up computation and improve the approximation guarantee) is to remove dominated objectives. An objective function F_i is dominated by another objective F_j, if $F_i(\mathcal{A}) \geq F_j(\mathcal{A})$ for all sets $\mathcal{A} \subseteq \mathcal{V}$. Hence, an F_i is dominated by F_j if an adversary can always reduce our score by choosing F_j instead of F_i. For example, when considering sensor failures or feature deletion (as discussed in Section 6.5), for two sets $\mathcal{B} \subseteq \mathcal{B}'$, the objective $F_{\mathcal{B}}$ is dominated by the objective $F_{\mathcal{B}'}$, i.e., the score decreases more if more sensors fail. Similarly, in the case of outbreak detection, some outbreak scenarios have much more impact on the network than others. Even though objective functions measuring the impact reductions F_i for scenarios $i \in \mathcal{I}$ might not be *exactly* dominated, they might be ε-dominated, i.e., $F_i(\mathcal{A}) \geq F_j(\mathcal{A}) - \varepsilon$ for some $\varepsilon > 0$ and all $\mathcal{A} \subseteq \mathcal{V}$. In such cases, these approximately dominated scenarios can be removed, incurring at most an error of ε in the quality of the approximate solution.

9.2 Constraint Generation

Another possible approach to reduce the number m of objective functions is constraint generation (*cf.,* Benders 1962). In this approach, one starts with an arbitrary single objective function, F_1. In iteration $j+1$, $(j \geq 1)$, after functions $F_1, \ldots, F_j$ have been considered, one searches for set $\mathcal{A}_j$ maximizing $\max_{\mathcal{A}} \min_{1 \leq i \leq j} F_j(\mathcal{A})$. Subsequently, one selects F_{j+1} minimizing $\min_i F_i(\mathcal{A}_j)$. The iteration terminates once F_{j+1} is contained in the already selected objectives $F_1, \ldots, F_j$. Another option is to terminate once the new objective F_{j+1} is ε-dominated by some objective F_i, $1 \leq i \leq j$. In this case, the approximate solution is guaranteed to incur at most an absolute error of ε as compared to the optimal solution.

In order to implement this constraint generation scheme, one must be able to efficiently solve problem $\min_i F_i(\mathcal{A}_j)$. In some settings, this problem might admit an efficient (perhaps approximate) solution. In many problems, such as the experimental design setting, one actually wants to perform well against an (uncountably) infinite set of possible objective functions, corresponding to parameters $\theta \in D$ in some (typically compact and convex set D). In such a setting, $\min_\theta F_\theta(\mathcal{A}_j)$ could potentially be (at least heuristically) solved using a numerical optimization approach such as a conjugate gradient method.

10 Related Work

10.1 Submodular Function Optimization

In their seminal work, Nemhauser et al. (1978) and Wolsey (1982) analyze the greedy algorithm for optimizing monotonic submodular functions. Lovász (1983) discusses the relationship between submodular functions and convexity. He also shows that under certain conditions, the minimum of two submodular functions remains submodular (and hence can be efficiently optimized using the greedy algorithm). The objective functions resulting from observation selection problems typically do not satisfy these properties, and, as we have shown, the greedy algorithm can perform arbitrarily badly. Fujito (2000) uses submodularity of truncated functions to find sets with partial submodular coverage; however, they do not consider the case of multiple objectives, which we address in this paper. Bar-Ilan et al.

(2001) consider covering problems for a generalization of submodular functions; they use a similar binary search technique combined with multiple applications of the greedy algorithm. Their approach does not apply to maximizing the minimum over a set of submodular functions. Golovin and Streeter (2008) present an algorithm for online maximization of a single submodular set function. An interesting question for future work would be to investigate whether our approach for maximizing the minimum over a collection of submodular functions can be generalized to an online setting as well.

A large part of the theory of optimizing submodular functions is concerned with *minimizing* instead of maximizing a single submodular function. Queyranne (1995) present the first algorithm for minimizing symmetric submodular functions; Iwata et al. (2001) and Schrijver (2000) present combinatorial algorithms for minimizing *arbitrary* (not necessarily symmetric) submodular functions.

10.2 Robust Discrete Optimization

Robust optimization of submodular functions is an instance of a robust discrete optimization problem. In such problems, the goal generally is to perform well with respect to a worst-case choice of evaluation scenario. Other instances of robust discrete problems have been studied by a number of authors. Kouvelis and Yu (1997) introduce several notions of robust discrete problems, presents hardness results and a class of robust problems that can be optimally solved. Averbakh (2001) shows that a class of robust optimization problems (selecting a k-element subset of elements of minimum cost) is solvable in polynomial time if the uncertain cost coefficients are contained in an interval, but NP-hard under an arbitrary (finite) set of adversarially chosen scenarios. Bertsimas and Sim (2003) proposes a class of robust mixed integer programs, accommodating uncertainty both in cost and data coefficients. They show that in certain cases (robust matching, spanning tree, etc.), the robust formulations are solvable in polynomial time if the non-robust problem instances are solvable in polynomial time. In the case of NP-hard but α-approximable non-robust problems, they show that the corresponding robust formulations also remain α-approximable. However, their results do not transfer to our setting of robust submodular optimization, since in this case, even though non-robust solutions are $(1 - 1/e)$ approximable, the non-robust formulation does not admit any approximation guarantees (*cf.,* Section 3).

10.3 Robust Methods in Statistics

10.3.1 Robust Experimental Design

Experimental design under parameter uncertainty has been studied in statistics; most of the earlier work is reviewed in the excellent survey of Chaloner and Verdinelli (1995). In the survey, the authors discuss Bayesian approaches to handling parameter uncertainty, as well as robust Bayesian (*cf.,* Berger 1984) approaches, which perform worst-case analyses over prior and likelihood functions. In experimental design, most approaches have focused on *locally* optimal designs, i.e., those selecting an optimal design based on a linearization around an initial parameter estimate, for reasons of computational tractability. In order to cope with uncertainty in the initial parameter estimates around which linearization is performed, heuristic techniques have been developed, such as the SDP based approach of

Flaherty et al. (2006), or a clustering heuristic described by Dror and Steinberg (2006). We are not aware of approaches which allow to find designs in the context of such parameter uncertainty that bear theoretical guarantees similar to the approach described in this paper.

10.3.2 Minimax Kriging

Minimizing the maximum predictive variance in Gaussian Process regression has been proposed as a design criterion by Burgess et al. (1981) and since then extensively used. (*cf.,* Sacks and Schiller, 1988; van Groenigen and Stein, 1998). To our knowledge, prior to this work, no algorithms with approximation guarantees are known for this criterion.

Several authors consider the problem of spatial prediction under unknown covariance parameters. Pilz et al. (1996) describes an approach for selecting – for a fixed set of observed sites – the Kriging estimate minimizing the maximum prediction error, where the worst-case over a fixed class of covariance functions is assumed. Wiens (2005) consider a similar setting but also addresses the design problem of choosing locations in order to minimize the mean squared prediction error against the worst-case covariance function. Algorithmically, Wiens (2005) use the simulated annealing algorithm described in Section 8.1 with 7 tuned parameter settings. Note that the SATURATE algorithm can be used in this context as well.

10.4 Sensor Placement and Facility Location

Carr et al. (2006) consider the problem of robust sensor placements in water distribution networks. They formulate Mixed Integer Programs for selecting sensor placements robust against uncertainty in adversarial strategies and in water demands. Due to computational complexity of Mixed Integer Programming, in their experiments, they used only small networks of at most 470 nodes. SATURATE can potentially be applied to handle uncertainty in demands as well, which is an interesting direction for future work. Watson et al. (2006) consider different notions of robustness in the context of water distribution networks, intended to remove some of the pessimistic assumptions of purely robust sensor placements. They develop integer programs, as well as heuristics, and apply them to networks of similar size as the one considered in this paper. Their local search heuristic performs a sequence of local moves similar to those performed by the simulated annealing algorithm considered in Section 8.3, and does not provide any theoretical guarantees.

Closely related to the adversarial outbreak detection problem is the *k-center* problem. In this problem, one is given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ along with a distance function defined over pairs of nodes in $\mathcal{V}$. The goal is to select a subset $\mathcal{A} \subseteq \mathcal{V}$ of size at most k, such that the maximum distance between any unselected node $s \in \mathcal{V} \setminus \mathcal{A}$ and its nearest center $s' \in \mathcal{A}$ is minimized. For this problem, Minieka (1970) discuss a technique reducing the solution of this problem to a sequence of set cover problems combined in a binary search, similar in spirit to SATURATE. However, they do not discuss any implications regarding approximation guarantees, and do not consider the case of arbitrary submodular functions. Mladenovic et al. (2003) presents a Tabu search heuristic for k-center, also without theoretical guarantees. Gonzalez (1985) and Hochbaum and Shmoys (1985) present a 2 approximation for the k-center problem in the case of symmetric distance functions satisfying the triangle inequality. Panigrahy and Vishwanathan (1998) present a $\log^*(n)$ approximation in the case of distance functions satisfying the asymmetric triangle inequality, which is shown to

be best possible by Chuzhoy et al. (2005). Chuzhoy et al. (2005) also show that even for bicriterion algorithms (such as SATURATE), k-center is $\log^*(n)$ hard to approximate, even if $\mathcal{O}(k)$ additional centers can be selected. Note that SATURATE can be used to solve k-center problems (without any requirements on symmetry or on the triangle inequality), hence the bicriterion hardness result of Chuzhoy et al. (2005) gives further evidence on the tightness of the guarantees described in Section 5.

Anthony et al. (2008) consider robust and stochastic notions of facility location problems (such as k-center and k-median, where, instead of the maximum distance the average distance is optimized). In contrast to the robust problems in this paper which want to select k elements to *maximize* the minimum value achieved by these k elements over the m scenarios, the problems in Anthony et al. (2008) try to select k "centers" in a metric space to *minimize* the maximum cost incurred over the m scenarios—where the cost is some function of the distances between non-selected vertices to the selected centers. For several such robust cost-minimization problems in cases where distances satisfy the symmetric triangle inequality, they present an algorithm that opens k "centers" and achieves an approximation ratio of $\mathcal{O}(\log n + \log m)$ (where n is the number of nodes in the graph, and m is the number of scenarios): this should be compared to the impossibility results for approximating robust value-maximization problems presented in this paper.

10.5 Relationship to Game Theory and Allocation Problems

The RSOS problem can be viewed as the problem of finding an optimal pure strategy for a zero-sum matrix game with player ordering. In this matrix game, the rows would correspond to the possible sensor placements, and the columns would correspond to the objective functions F_i. The entry for cell $(\mathcal{A}, F_i)$ is our payoff $F_i(\mathcal{A})$. In the RSOS problem, we want to select a row of the matrix, our adversary selects a column F_i (knowing our choice $\mathcal{A}$, hence the player ordering) minimizing our score $F_i(\mathcal{A})$. A very related class of game theoretic problems are *allocation problems*. In these problems, one is typically given a set $\mathcal{V}$ of objects, and the goal is to allocate the objects to m agents (bidders), each of whom has a (potentially different) valuation function $F_i(\mathcal{A}_i)$ defined over subsets of received items $\mathcal{A}_i$. The problem of finding the best such allocation (partition) is NP-hard, but recently, several approximation algorithms have been proposed. The allocation problem most similar to the RSOS problem is

$$\pi^* = \operatorname*{argmax}_{\text{partition } \pi=(\mathcal{A}_1,\ldots,\mathcal{A}_m)} \min_i F_i(\mathcal{A}_i).$$

The main difference is that in the allocation problem, the full set $\mathcal{V}$ is partitioned into subsets $\mathcal{A}_1, \ldots, \mathcal{A}_m$, and the functions F_i are evaluated on the respective subset $\mathcal{A}_i$ each. In the case of *additive* objective functions F_i, Asadpour and Saberi (2007) provide an $\mathcal{O}(\sqrt{k}\log^3 k)$ approximation algorithm. In the case of the function being *subadditive* (which is implied by, and is more general than, submodularity), Ponnuswami and Khot (2007) present an $\mathcal{O}(2k-1)$ approximation algorithm. For settings where the *sum* of the valuations is optimized, i.e.,

$$\pi^* = \operatorname*{argmax}_{\text{partition } \pi=(\mathcal{A}_1,\ldots,\mathcal{A}_m)} \sum_i F_i(\mathcal{A}_i),$$

Feige (2006) develop a randomized 2-approximation for subadditive and $1 - 1/e$ approximation for submodular valuation functions.

The problem of trading off safety (i.e., improvements in worst-case scores) and average case performance has been studied by several authors. Johanson et al. (2007) consider the problem of opponent modeling in games, and develop an algorithm which can exploit opponents which it can accurately model, and falls back to a safe (Nash) strategy in case the models do not capture the opponents behavior. Their algorithm has a tradeoff parameter which controls the eagerness of exploiting, and they present Pareto-curves similar to those presented in Section 7.5. However, their approach does not apply to our robust submodular observation selection setting. Watson et al. (2006) consider different optimization problem formulations allowing to control risk in the water distribution network monitoring application, but they only present heuristic algorithms without guarantees for coping with large networks.

10.6 Relationship to Machine Learning

Submodular function optimization has found increasing use in machine learning. The algorithm of Queyranne (1995) for minimizing symmetric submodular functions has been used for learning graphical models by Narasimhan and Bilmes (2004) and for clustering by Narasimhan et al. (2005). We are not aware of any work on optimizing the minimum over a collection of submodular functions.

Observation selection approaches have been used in the context of active learning (*cf.,* Sollich, 1996; Freund et al., 1997; Axelrod et al., 2001; MacKay, 1992; Cohn, 1994). Test point selection has been used to minimize average predictive variance in Gaussian Processes regression by Seo et al. (2000), and to speed up Gaussian Process inference by Seeger et al. (2003); Lawrence et al. (2003). In these approaches, the sequential setting is considered, where previous measurements are taken into account when deciding on the next observation to make. The extension of the robust techniques discussed in this paper, which address the a priori selection problem (i.e., observations are selected before measurements are obtained), to the sequential setting is an important direction for future research.

Balcan et al. (2006) consider the problem of active learning in the presence of *adversarial* noise. While their method is very different, our results potentially generalize to active learning settings, since, as Hoi et al. (2006) show, certain active learning objectives are (approximately) submodular.

Price and Messinger (2005) consider the problem of constructing recommendation sets, and show that this problem is an instance of a k-median problem (*cf.,* Section 10.4). The analogue of the k-center problem in the preference set construction would be to construct a preference set which maximizes the utility of displayed items under worst-case instantiation of the parameters. This analogue seems natural, and an interesting direction for future work would be to explore the use of SATURATE in the recommendation set context.

10.7 Relationship to Previous Work of the Authors

A previous version of this paper appeared in (Krause et al., 2007a). The present version is significantly extended, providing new theoretical analyses (described in Section 7, Section 9), new examples demonstrating the generality of the observation selection problem (Section 6)

and additional empirical results (Section 8). In previous work, the authors demonstrated that several important observation selection objectives are submodular (Krause et al., 2007b; Leskovec et al., 2007; Krause and Guestrin, 2005, 2007a). Krause et al. (2006) consider the problem of optimizing the placement of a network of wireless sensors. In this context, the chosen locations must be both informative and communicate well, constraining the chosen locations not to be too far apart. Singh et al. (2007); Meliou et al. (2007) consider the problem of planning informative paths for multiple robots, where the informativeness is modeled using a submodular objective function, and a constraint on path lengths connecting the locations is specified. In the context of such more complex (communication and path) constraints – similarly to the robust setting – the greedy algorithm can fail arbitrarily badly, and more complex algorithms have to be developed. Using the techniques described in Section 7.4, both approaches can be made robust with respect to a worst-case submodular function.

11 Conclusions

In this paper, we considered the RSOS problem of robustly selecting observations which are informative with respect to a worst-case submodular objective function. We demonstrated the generality of this problem, and showed how it encompasses the problem of sensor placements which minimize the maximum posterior variance in Gaussian Process regression, variable selection under parameter uncertainty, robust experimental design, and detecting events spreading over graphs, even in the case of adversarial sensor failures. In each of these settings, the individual objectives are submodular and can be approximated well using, e.g., the greedy algorithm; the robust objective, however, is not submodular.

We proved that there cannot exist any approximation algorithm for the robust optimization problem if the constraint on the observation set size must be exactly met, unless P = NP. Consequently, we presented an efficient approximation algorithm, SATURATE, which finds observation sets which are guaranteed to be least as informative as the optimal solution, and only logarithmically more expensive. In a strong sense, this guarantee is the best possible under reasonable complexity theoretic assumptions.

We provided several extensions to our methodology, accommodating more complex cost functions (non-uniform observation costs, communication and path costs). Additionally, we described how a compromise between worst-case and average-case performance can be achieved. We also discussed several approaches for reducing the number of objective functions, improving both running times and theoretical guarantees.

We extensively evaluated our algorithm on several real-world problems. For Gaussian Process regression, for example, we showed that SATURATE compares favorably to state-of-the-art heuristics, while being simpler, faster, and providing theoretical guarantees. For robust experimental design, SATURATE performs favorably compared to SDP based approaches. We believe that the ideas developed in this paper will help the development of robust monitoring systems and provide new insights for adapting machine learning algorithms to cope with adversarial environments.

Acknowledgements

We would like to thank Michael Bowling for helpful discussions. This work was partially supported by NSF Grants No. CNS-0509383, CNS-0625518, CCF-0448095, CCF-0729022, and a gift from Intel. Anupam Gupta and Carlos Guestrin were partly supported by Alfred P. Sloan Fellowships, Carlos Guestrin by an IBM Faculty Fellowship and Andreas Krause by a Microsoft Research Graduate Fellowship.

References

Barbara M. Anthony, Vineet Goyal, Anupam Gupta, and Viswanath Nagarajan. A plant location guide for the unsure. In *Submitted to SODA*, 2008.

Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *STOC*, pages 114–121, New York, NY, USA, 2007. ACM Press. ISBN 978-1-59593-631-8. doi: http://doi.acm.org/10.1145/1250790.1250808.

Igor Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90:263–272, 2001.

S. Axelrod, S. Fine, R. Gilad-Bachrach, R. Mendelson, and N. Tishby. The information of observations and application for active learning with uncertainty. Technical report, Jerusalem: Leibniz Center, Hebrew University, 2001.

M. F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, 2006.

J. Bar-Ilan, G. Kortsarz, and D. Peleg. Generalized submodular cover problems and applications. *Theoretical Computer Science*, 250(1-2):179–200, January 2001.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

J. Berger. *Robustness of Bayesian Analyses*, chapter The robust Bayesian viewpoint, page 63144. North-Holland, 1984.

Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71, 2003.

Avrim Blum, Shuchi Chawla, David R. Karger, Terran Lane, Adam Meyerson, and Maria Minkoff. Approximation algorithms for orienteering and discounted-reward tsp. In *FOCS*, page 46, 2003.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge UP, March 2004.

T.M. Burgess, R. Webster, and A.B. McBratney. Optimal interpolation and isarithmic mapping of soil properties. iv. sampling strategy. *Journal of Soil Science*, 32:643–659, 1981.

R. D. Carr, H. J. Greenberg, W. E. Hart, G. Konjevod, E. Lauer, H. Lin, T. Morrison, and C. A. Phillips. Robust optimization of contaminant sensor placement for community water systems. *Mathematical Programming Series B*, 107:337–356, 2006.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, Aug. 1995. ISSN 08834237.

J. Chuzhoy, S. Guha, E. Halperin, S. Khanna, G. Kortsarz, R. Krauthgamer, and J. Naor. Asymmetric k-center is $\log^* n$-hard to approximate. *Journal of the ACM*, 52(4):538–551, 2005.

D. A. Cohn. Neural network exploration using optimal experiment design. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 679–686. Morgan Kaufmann Publishers, Inc., 1994.

N. A. C. Cressie. *Statistics for Spatial Data*. Wiley, 1991.

A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *Manuscript*, 2007.

H. A. Dror and D. M. Steinberg. Robust experimental design for multivariate generalized linear models. *Technometrics*, 48(4):520–529, 2006.

U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4), 1998.

Uriel Feige. On maximizing welfare when utility functions are subadditive. In *STOC*, 2006.

P. Flaherty, M. Jordan, and A. Arkin. Robust design of biological experiments. In *NIPS*, 2006.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

T. Fujito. Approximation algorithms for submodular set cover with applications. *TIEICE*, 2000. URL `citeseer.ist.psu.edu/article/fujito00approximation.html`.

Amir Globerson and Sam Roweis. Nightmare at test time: Robust learning by feature deletion. In *ICML*, 2006.

Daniel Golovin and Matthew Streeter. Online algorithms for maximizing submodular set functions. In *Submitted to SODA*, 2008.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38(2-3):293–306, 1985. ISSN 0304-3975.

C. Guestrin, A. Krause, and A. Singh. Near-optimal sensor placements in Gaussian processes. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML)*, 2005.

T. C. Harmon, R. F. Ambrose, R. M. Gilbert, J. C. Fisher, M. Stealey, and W. J. Kaiser. High resolution river hydraulic and water quality characterization using rapidly deployable networked infomechanical systems (nims rd). Technical Report 60, CENS, 2006.

D. Hochbaum and D. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *ICML*, 2006.

Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

Michael Johanson, Martin Zinkevich, and Michael Bowling. Computing robust counter-strategies. In *NIPS*, 2007.

David S. Johnson, Maria Minkoff, and Stephen Phillips. The prize collecting steiner tree problem: theory and practice. In *SODA*, 2000.

A. K. Kelmans and B. N. Kimelfeld. Multiplicative submodularity of a matrix's principal minor as a function of the set of its rows and some combinatorial applications. *Discrete Mathematics*, 44(1):113–116, 1980. doi: http://dx.doi.org/10.1016/0012-365X(83)90011-0.

P. Kouvelis and G. Yu. *Robust Discrete Optimization and its Applications*. Kluwer Academic Publishers, 1997.

A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *UAI*, 2005.

A. Krause and C. Guestrin. Near-optimal observation selection using submodular functions. In *AAAI Nectar track*, 2007a.

A. Krause and C. Guestrin. Nonmyopic active learning of gaussian processes: An exploration—exploitation approach. In *ICML*, 2007b.

A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the Fifth International Symposium on Information Processing in Sensor Networks (IPSN)*, 2006.

A. Krause, B. McMahan, C. Guestrin, and A. Gupta. Selecting observations against adversarial objectives. In *NIPS*, 2007a.

A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. In *To appear in the JMLR*, 2007b.

Gilbert Laporte and Silvano Martello. The selective travelling salesman problem. *Disc. App. Math*, 26:193–207, 1990.

N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems (NIPS) 16*, 2003.

J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.

L. Lovász. Submodular functions and convexity. *Mathematical Programming - State of the Art*, pages 235–257, 1983.

D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, 1990.

Alexandra Meliou, Andreas Krause, Carlos Guestrin, and Joseph M. Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *AAAI*, 2007.

E. Minieka. The m-center problem. *SIAM Rev*, 12(1):138–139, 1970.

Nenad Mladenovic, Martine Labbé, and Pierre Hansen. Solving the p-center problem with tabu search and variable neighborhood search. *Networks*, 42(1):48–64, 2003.

M. Narasimhan and J. Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. In *Advances in Neural Information Processing Systems (NIPS) 19*, 2006.

Mukund Narasimhan and Jeff Bilmes. Pac-learning bounded tree-width graphical models. In *Uncertainty in Artificial Intelligence*, 2004.

Mukund Narasimhan, Nebojsa Jojic, and Jeff Bilmes. Q-clustering. In *NIPS*, 2005.

G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

A. Ostfeld, J. G. Uber, and E. Salomons. Battle of water sensor networks: A design challenge for engineers and algorithms. In *8th Symposium on Water Distribution Systems Analysis*, 2006.

Avi Ostfeld, James G. Uber, Elad Salomons, Jonathan W. Berry, William E. Hart, Cindy A. Phillips, Jean-Paul Watson, Gianluca Dorini, Philip Jonkergouw, Zoran Kapelan, Francesco di Pierro, Soon-Thiam Khu, Dragan Savic, Demetrios Eliades, Marios Polycarpou, Santosh R. Ghimire, Brian D. Barkdoll, Roberto Gueli, Jinhui J. Huang, Edward A. McBean, William James, Andreas Krause, Jure Leskovec, Shannon Isovitsch, Jianhua, Carlos Guestrin, Jeanne VanBriesen, Mitchell Small, Paul Fischbeck, Ami Preis, Marco Propato, Olivier Piller, Gary B. Trachtman, Zheng Yi Wu, and Tom Walski. The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms. *submitted to Journal of Water Resources Planning and Management*, 2008.

Rina Panigrahy and Sundar Vishwanathan. An $\mathcal{O}(\log^* n)$ approximation algorithm for the asymmetric p-center problem. *Journal of Algorithms*, 27(2):259–268, 1998.

C. H. Papadimitriou and M. Yannakakis. The complexity of tradeoffs, and optimal access of web sources. In *FOCS*, 2000.

J. Pilz, G. Spoeck, and M. G. Schimek. *Geostatistics Wollongong*, volume 1, chapter Taking account of uncertainty in spatial covariance estimation, pages 302–313. Kluwer, 1996.

Ashok Kumar Ponnuswami and Subhash Khot. Approximation algorithms for the max-min allocation problem. In *APPROX*, 2007.

Robert Price and Paul R. Messinger. Optimal recommendation sets: Covering uncertainty over user preferences. In *AAAI*, 2005.

Maurice Queyranne. A combinatorial algorithm for minimizing symmetric submodular functions. In *SODA*, 1995.

S. Rajagopalan and V.V. Vazirani. Primaldual rnc approximation algorithms for set cover and covering integer programs. *SIAM Journal on Computing*, 28(2):525–540, 1998.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Process for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2006.

T. G. Robertazzi and S. C. Schwartz. An accelerated sequential algorithm for producing D-optimal designs. *SIAM Journal of Scientific and Statistical Computing*, 10(2):341–358, March 1989.

L. A. Rossman. The epanet programmer's toolkit for analysis of water distribution systems. In *Annual Water Resources Planning and Management Conference*, 1999.

J. Sacks and S. Schiller. *Statistical Decision Theory and Related Topics IV, Vol. 2*. Springer, 1988.

Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Combin. Theory Ser. B*, 80(2):346–355, 2000. ISSN 0095-8956.

M. Seeger, C. K. I. Williams, and N. D. Lawrence. Fast forward selection to speed up sparse gaussian process regression. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2003.

S. Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: Active data selection and test point rejection. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 241–246, 2000.

A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin. Efficient planning of informative paths for multiple robots. In *IJCAI*, 2007.

P. Sollich. Learning from minimum entropy queries in a large committee machine. *Physical Review E*, 53:R2060–R2063, 1996.

J.W. van Groenigen and A. Stein. Constrained optimization of spatial sampling using continuous simulated annealing. *J. Environ. Qual.*, 27:1078–1086, 1998.

Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2003.

J. Watson, W. E. Hart, and R. Murray. Formulation and optimization of robust sensor placement problems for contaminant warning systems. In *Water Distribution System Symposium*, 2006.

M. Widmann and C. S. Bretherton. 50 km resolution daily precipitation for the pacific northwest. http://www.jisao.washington.edu/data_sets/widmann/, May 1999.

D. P. Wiens. Robustness in spatial studies ii: minimax design. *Environmetrics*, 16:205–217, 2005.

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

A Proofs

[Theorem 3] Consider a hitting set instance with m subsets $\mathcal{S}_i \subseteq \mathcal{V}$ on a ground set $\mathcal{V}$. Our task is to select a set $\mathcal{A} \subseteq \mathcal{V}$ with which intersects all sets $\mathcal{S}_i$, and such that $|\mathcal{A}| = k$ is as small as possible. For each set $\mathcal{S}_i$, define a function F_i such that $F_i(\mathcal{A}) = 1$ if $\mathcal{A}$ intersects $\mathcal{S}_i$, and 0 otherwise. It can be seen that F_i is clearly monotonic. F_i is also submodular, since for $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $x \in \mathcal{V} \setminus \mathcal{B}$, if $F_i(\mathcal{B}) = 0$ and $F_i(\mathcal{B} \cup \{x\}) = 1$, then it $x \in \mathcal{S}_i$, hence $F_i(\mathcal{A} \cup \{x\}) = 1$ and $F_i(\mathcal{A}) = 0$. Now assume the optimal hitting set $\mathcal{A}^*$ is of size k. Hence $\min_i F_i(\mathcal{A}^*) = 1$. If there were an algorithm for solving Problem (2.2) with approximation guarantee $\alpha(n)$ it would select a set $\mathcal{A}'$ of size $|\mathcal{A}'| \leq k$ with $\min_i F_i(\mathcal{A}') \geq \alpha(n) \min_i F_i(\mathcal{A}^*) = \alpha(n) > 0$. But $\min_i F_i(\mathcal{A}') > 0$ implies $\min_i F_i(\mathcal{A}') = 1$, hence $\mathcal{A}'$ would be a hitting set. Hence, this approximation algorithm would be able to decide, whether there exists a hitting set of size k, contradicting the NP-hardness of the hitting set problem (Feige, 1998).

[Theorem 5] Lemma 4 proves that during each of the iterations of the saturation algorithm it holds that $\min_i F_i(\mathcal{A}^*) \leq c_{\max}$, where $\mathcal{A}^*$ is an optimal solution. Furthermore, it holds that $\min_i F_i(\mathcal{A}_{best}) \geq c_{\min}$, and $\mathcal{A}_{best} \leq \alpha k$. Since the F_i are integral, if $c_{\max} - c_{\min} < \frac{1}{m}$ then it must hold that $\min_i F_i(\mathcal{A}_{best}) \geq \min_i F_i(\mathcal{A}^*)$ as claimed by Theorem 5.

For the running time, since at the first iteration, $c_{\max} - c_{\min} \leq \frac{1}{m} \sum_i F_i(\mathcal{V})$, and $c_{\max} - c_{\min}$ is halved during each iteration, it follows that after $1 + \lceil \log_2 \sum_i F_i(\mathcal{V}) \rceil$ iterations, $c_{\max} - c_{\min} < \frac{1}{m}$, at which point the algorithm terminates. During each iteration, Algorithm 1 is invoked once, which requires $\mathcal{O}(|\mathcal{V}|^2 m)$ function evaluations.

[Theorem 6] We use the same hitting set construction as in Theorem 3. If there were an algorithm for selecting a set $\mathcal{A}'$ of size $|\mathcal{A}'| \leq \beta k$ with $\min_i F_i(\mathcal{A}') = 1$, and $\beta = o(\alpha)$, then we would have an approximation algorithm for hitting set with guarantee $o(\log m)$ which would imply NP $\subseteq$ DTIME($n^{\log \log n}$) (Feige, 1998).

[Theorem 8] The proof is analogous to the proof of Theorem 5. The approximation guarantee α is established by noticing that the greedy algorithm is applied to the modified (integral) objective

$$\overline{F}_{c_{wc}, c_{ac}}(\mathcal{A}) = \sum_i \min\{F_i(\mathcal{A}), c_{wc}\} + \min\left\{\sum_i F_i(\mathcal{A}), m c_{ac}\right\}.$$

The guarantee α is obtained from the analysis of the greedy submodular coverage algorithm of Wolsey (1982), similar to Lemma 4. Approximate Pareto-optimality follows directly from Pareto-optimality of any solution to (7.3).

B Scalarization

In the scalarized objective version of the problem we optimize the objective

$$F_\lambda(\mathcal{A}) = \lambda F_{wc}(\mathcal{A}) + (1-\lambda)F_{ac}(\mathcal{A})$$

for a fixed parameter $\lambda \in [0,1]$. The scalarization parameter λ can be though of as a prior: with λ probability we expect to face an adversarial objective, while with probability $1-\lambda$ we expect an objective drawn from the uniform distribution[4] F_{ac}.

It turns out that this formulation is equivalent to facing an adversary who, rather than being able to pick any F_i arbitrarily, instead chooses a probability distribution on the F_i from some convex set of probability distributions $\mathcal{P}_{\text{wc}}$. If $\mathcal{P}_{\text{wc}} = \Delta(m)$, the set of all possible probability distributions on the m functions F_i, then we recover the fully adversarial problem as the adversary can choose a probability distribution that puts probability 1 on the best response to our chosen sensor placements:

$$F_{wc}(\mathcal{A}) = \min_{p \in \Delta(m)} \sum_i p_i F_i(\mathcal{A}) \tag{B.1}$$

The scalarized multi-criterion objective F_λ corresponds to a more interesting convex set,

$$\mathcal{P}^\lambda_{\text{wc}} = \left\{ q \mid q_i = \lambda p_i + \frac{1-\lambda}{m} \text{ for some } p \in \Delta(m) \right\},$$

which can be derived as follows:

$$\begin{aligned}
F_\lambda(\mathcal{A}) &= \lambda F_{wc}(\mathcal{A}) + (1-\lambda)F_{ac}(\mathcal{A}) \\
&= \lambda \min_{p \in \Delta(m)} \sum_i p_i F_i(\mathcal{A}) + (1-\lambda) \sum_i \frac{1}{m} F_i(\mathcal{A}) && \text{by (B.1)} \\
&= \min_{p \in \Delta(m)} \sum_i \left(\lambda p_i + \frac{1-\lambda}{m} \vec{1} \right) F_i(\mathcal{A}) \\
&= \min_{q \in \mathcal{P}^\lambda_{\text{wc}}} \sum_i q_i F_i(\mathcal{A}).
\end{aligned}$$

It is straightforward to show $\mathcal{P}^\lambda_{\text{wc}} = \{q \mid q_i \geq (1-\lambda)/m,\ \sum_i q_i = 1\}$, and so the scalarized multi-criterion objective is equivalent to solving the adversarial problem where the adversary must put at least probability $(1-\lambda)/m$ on every scenario.

In order to solve this problem, we recast F_λ again as

$$F_\lambda(\mathcal{A}) = \sum_i \left(\lambda F_i(\mathcal{A}) + (1-\lambda)F_{ac}(\mathcal{A}) \right).$$

[4] In fact, this argument easily generalizes to an arbitrary fixed distribution on the F_i

The functions $F_i^\lambda = \lambda F_i(\mathcal{A}) + (1-\lambda)F_{ac}(\mathcal{A})$ are submodular, and so we can solve the scalarized version of the multi-criterion optimization by running an unmodified SATURATE algorithm on the set of objective functions F_i^λ.

In fact, this algorithmic technique can be extended to an arbitrary convex set of probability distributions $\mathcal{P}_{\text{wc}}$ with a finite number of extreme points. An extreme point of $\mathcal{P}_{\text{wc}}$ is a distribution that cannot be expressed as a convex combination of other distributions in $\mathcal{P}_{\text{wc}}$. Since the adversary knows our choice $\mathcal{A}$, the optimization over $\mathcal{P}_{\text{wc}}$ to find the best-response is the optimization of a linear objective over a bounded convex set, and so there always exists an extreme point that is optimal.

An extreme point q is a distribution over the original F_i, and in fact corresponds exactly to the submodular expected cost function

$$\sum_i q_i F_i(\mathcal{A}).$$

Hence, if we have a finite number of extreme points we can solve the set-selection problem against an adversary constrained to play from $\mathcal{P}_{\text{wc}}$ by running SATURATE on the set of derived expected-cost functions corresponding to the extreme points.

C Examples of submodular functions

In this Section, we review several examples of submodular functions.

Cardinality of union (set cover) Perhaps the most well-known example is the *cardinality of union*. Consider an application where we want to place a set of cameras covering an area. With each camera, we associate a fixed field of view (e.g., a cone). Our goal is to place the cameras such that as much area as possible is covered. More formally, we can define a finite ground set $\mathcal{S}$ (consisting, e.g., of a discretization of the space to be covered into a finite number of locations), and a collection of subsets $\mathcal{W}_1, \ldots, \mathcal{W}_n \subseteq \mathcal{S}$ (each corresponding, e.g., to the view field of a camera when placed at one of n possible locations). The function

$$F(\mathcal{A}) = \left| \bigcup_{i \in \mathcal{A}} \mathcal{W}_i \right|,$$

(which quantifies the total number locations of $\mathcal{S}$ covered if cameras are placed at locations $\mathcal{A}$) is a normalized, monotonic and submodular function over the set $\mathcal{V} = \{1, \ldots, n\}$. Interesting extensions (preserving submodularity) allow different weights for the elements of $\mathcal{S}$ (e.g., some locations are more important than others), or covering elements of $\mathcal{S}$ multiple times (set multi-cover, *cf.,* Rajagopalan and Vazirani 1998, formalizing, e.g., a setting where we want to place the cameras with overlapping view fields such that each location is covered a specified number of times).

Information theoretic objectives Consider an application, where we want to diagnose a failure of a complex system, by performing a number of tests. We can model this problem by using a set of discrete random variables $\mathcal{X}_\mathcal{V} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ indexed by $\mathcal{V} = \{1, \ldots, n\}$, which model both the hidden state of the system and the outcomes of the diagnostic tests.

In such a setting, the joint entropy $F(\mathcal{A}) = H(\mathcal{X}_\mathcal{A})$ of subsets $\mathcal{A} \subseteq \mathcal{V}$ is normalized monotonic and submodular (Kelmans and Kimelfeld, 1980). In our diagnostic example, we are interested in selecting a subset of variables which are maximally informative with respect to a set of target variables $\mathcal{X}_\mathcal{U}$, modeling the hidden state of the system. Krause and Guestrin (2005) show that under certain conditional independence assumptions, the information gain $F(\mathcal{A}) = I(\mathcal{X}_\mathcal{A}; \mathcal{X}_\mathcal{U}) = H(\mathcal{X}_\mathcal{U}) - H(\mathcal{X}_\mathcal{U} \mid \mathcal{X}_\mathcal{A})$ is normalized, monotonic and submodular in $\mathcal{A}$. Other applications include tracking, feature selection etc.

Another important class of observation selection problems arises in the context of monitoring continuous, typically spatial phenomena (such as the temperature in a building). Such phenomena can often be modeled using Gaussian Processes (*cf.,* Cressie, 1991). Krause et al. (2007b) show that in the case of Gaussian Processes, the mutual information $F(\mathcal{A}) = I(\mathcal{X}_\mathcal{A}; \mathcal{X}_{\mathcal{V}\setminus\mathcal{A}})$ (i.e., reduction in uncertainty about the remaining unobserved locations $\mathcal{V} \setminus \mathcal{A}$) is normalized, submodular and approximately monotonic.

Variance reduction In continuous models such as Gaussian Processes, instead of using entropy to quantify the predictive uncertainty, other loss functions such as the predictive variance can be used. Das and Kempe (2007) show that under certain conditions, the variance reduction $F(\mathcal{A}) = \mathrm{Var}(\mathcal{X}_i) - \mathrm{Var}(\mathcal{X}_i \mid \mathcal{X}_\mathcal{A})$ of a fixed random variable $\mathcal{X}_i$ given observations $\mathcal{X}_\mathcal{A}$ is normalized, monotonic and submodular.