Journal of Machine Learning Research 9 (2008) 235-284 Submitted 9/06; Revised 9/07; Published 2/08

Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies

Andreas Krause KRAUSEA@CS.CMU.EDU
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Ajit Singh AJIT@CS.CMU.EDU
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213

Carlos Guestrin GUESTRIN@CS.CMU.EDU
Computer Science Department and Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213

Editor: Chris Williams

Abstract

When monitoring spatial phenomena, which can often be modeled as Gaussian processes (GPs), choosing sensor locations is a fundamental task. There are several common strategies to address this task, for example, geometry or disk models, placing sensors at the points of highest entropy (variance) in the GP model, and A-, D-, or E-optimal design. In this paper, we tackle the combinatorial optimization problem of maximizing the *mutual information* between the chosen locations and the locations which are not selected. We prove that the problem of finding the configuration that maximizes mutual information is NP-complete. To address this issue, we describe a polynomial-time approximation that is within $(1-1/e)$ of the optimum by exploiting the *submodularity* of mutual information. We also show how submodularity can be used to obtain online bounds, and design branch and bound search procedures. We then extend our algorithm to exploit lazy evaluations and local structure in the GP, yielding significant speedups. We also extend our approach to find placements which are robust against node failures and uncertainties in the model. These extensions are again associated with rigorous theoretical approximation guarantees, exploiting the submodularity of the objective function. We demonstrate the advantages of our approach towards optimizing mutual information in a very extensive empirical study on two real-world data sets.

Keywords: Gaussian processes, experimental design, active learning, spatial learning; sensor networks

1. Introduction

When monitoring spatial phenomena, such as temperatures in an indoor environment as shown in Figure 1(a), using a limited number of sensing devices, deciding where to place the sensors is a fundamental task. One approach is to assume that sensors have a fixed sensing radius and to solve the task as an instance of the art-gallery problem (cf. Hochbaum and Maas, 1985; Gonzalez-Banos and Latombe, 2001). In practice, however, this geometric assumption is too strong; sensors make noisy measurements about the nearby environment, and this "sensing area" is not usually characterized by a regular disk, as illustrated by the temperature correlations in Figure 1(b). In addition, note that correlations can be both positive and negative, as shown in Figure 1(c), which again is not well-characterized by a disk model. Fundamentally, the notion that a single sensor needs to predict values in a nearby region is too strong. Often, correlations may be too weak to enable prediction from a single sensor. In other settings, a location may be "too far" from existing sensors to enable good prediction if we only consider one of them, but combining data from multiple sensors we can obtain accurate predictions. This notion of combination of data from multiple sensors in complex spaces is not easily characterized by existing geometric models.

An alternative approach from spatial statistics (Cressie, 1991; Caselton and Zidek, 1984), making weaker assumptions than the geometric approach, is to use a pilot deployment or expert knowledge to learn a *Gaussian process* (GP) model for the phenomena, a non-parametric generalization of linear regression that allows for the representation of uncertainty about predictions made over the sensed field. We can use data from a pilot study or expert knowledge to learn the (hyper-)parameters of this GP. The learned GP model can then be used to predict the effect of placing sensors at particular locations, and thus optimize their positions.[1]

Given a GP model, many criteria have been proposed for characterizing the quality of placements, including placing sensors at the points of highest entropy (variance) in the GP model, and A-, D-, or E-optimal design, and mutual information (cf. Shewry and Wynn, 1987; Caselton and Zidek, 1984; Cressie, 1991; Zhu and Stein, 2006; Zimmerman, 2006). A typical sensor placement technique is to greedily add sensors where uncertainty about the phenomena is highest, that is, the highest entropy location of the GP (Cressie, 1991; Shewry and Wynn, 1987). Unfortunately, this criterion suffers from a significant flaw: entropy is an *indirect* criterion, not considering the prediction quality of the selected placements. The highest entropy set, that is, the sensors that are most uncertain about each other's measurements, is usually characterized by sensor locations that are as far as possible from each other. Thus, the entropy criterion tends to place sensors along the borders of the area of interest (Ramakrishnan et al., 2005), for example, Figure 4. Since a sensor usually provides information about the area around it, a sensor on the boundary "wastes" sensed information.

An alternative criterion, proposed by Caselton and Zidek (1984), *mutual information*, seeks to find sensor placements that are most informative about unsensed locations. This optimization criterion *directly* measures the effect of sensor placements on the posterior uncertainty of the GP. In this paper, we consider the combinatorial optimization problem of selecting placements which maximize this criterion. We first prove that maximizing mutual information is an NP-complete problem. Then, by exploiting the fact that mutual information is a *submodular* function (cf. Nemhauser et al., 1978), we design the first approximation algorithm that guarantees a *constant-factor approximation* of the best set of sensor locations in polynomial time. To the best of our knowledge, no such guarantee exists for any other GP-based sensor placement approach, and for any other criterion. This guarantee

[1] This initial GP is, of course, a rough model, and a sensor placement strategy can be viewed as an inner-loop step for an *active learning* algorithm (MacKay, 2003). Alternatively, if we can characterize the uncertainty about the parameters of the model, we can explicitly optimize the placements over possible models (Zidek et al., 2000; Zimmerman, 2006; Zhu and Stein, 2006).

holds both for placing a fixed number of sensors, and in the case where each sensor location can have a different cost.

Though polynomial, the complexity of our basic algorithm is relatively high—$O(kn^4)$ to select k out of n possible sensor locations. We address this problem in two ways: First, we develop a lazy evaluation technique that exploits submodularity to reduce significantly the number of sensor locations that need to be checked, thus speeding up computation. Second, we show that if we exploit locality in sensing areas by trimming low covariance entries, we reduce the complexity to $O(kn)$.

We furthermore show, how the submodularity of mutual information can be used to derive tight online bounds on the solutions obtained by any algorithm. Thus, if an algorithm performs better than our simple proposed approach, our analysis can be used to bound how far the solution obtained by this alternative approach is from the optimal solution. Submodularity and these online bounds also allow us to formulate a mixed integer programming approach to compute the optimal solution using Branch and Bound. Finally, we show how mutual information can be made robust against node failures and model uncertainty, and how submodularity can again be exploited in these settings.

We provide a very extensive experimental evaluation, showing that data-driven placements outperform placements based on geometric considerations only. We also show that the *mutual information* criterion leads to improved prediction accuracies with a reduced number of sensors compared to several more commonly considered experimental design criteria, such as an entropy-based criterion, and A-optimal, D-optimal and E-optimal design criteria.

In summary, our main contributions are:

- We tackle the problem of maximizing the information-theoretic *mutual information* criterion of Caselton and Zidek (1984) for optimizing sensor placements, empirically demonstrating its advantages over more commonly used criteria.

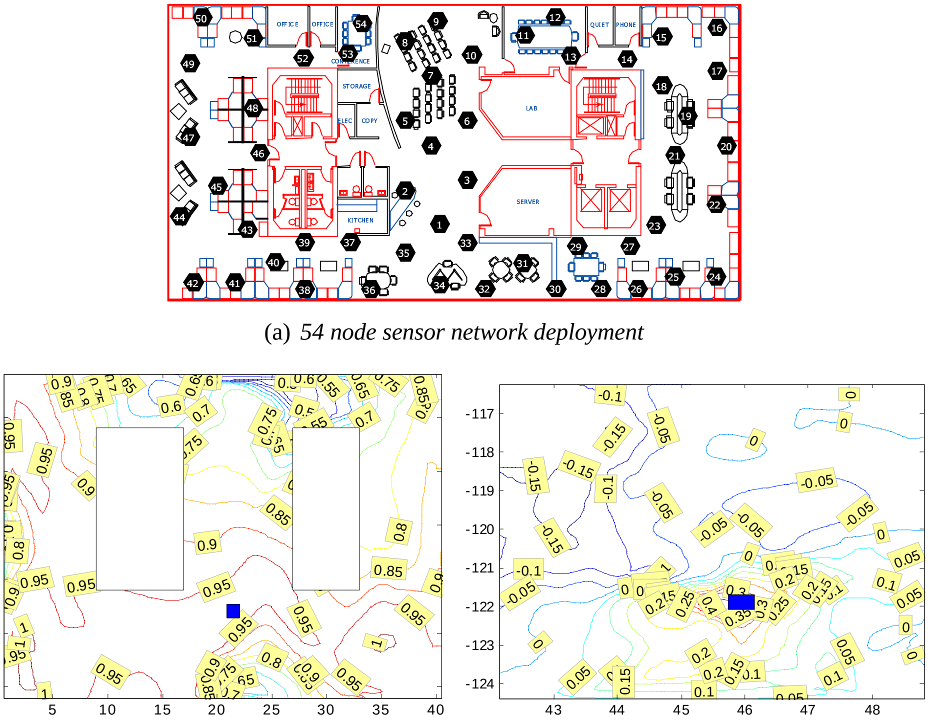
- Even though we prove NP-hardness of the optimization problem, we present a polynomial time approximation algorithm with constant factor approximation guarantee, by exploiting *submodularity*. To the best of our knowledge, no such guarantee exists for any other GP-based sensor placement approach, and for any other criterion.

- We also show that submodularity provides online bounds for the quality of our solution, which can be used in the development of efficient branch-and-bound search techniques, or to bound the quality of the solutions obtained by other algorithms.

- We provide two practical techniques that significantly speed up the algorithm, and prove that they have no or minimal effect on the quality of the answer.

- We extend our analysis of mutual information to provide theoretical guarantees for placements that are robust against failures of nodes and uncertainties in the model.

- Extensive empirical evaluation of our methods on several real-world sensor placement problems and comparisons with several classical design criteria.



(a) *54 node sensor network deployment*

(b) *Temperature correlations*

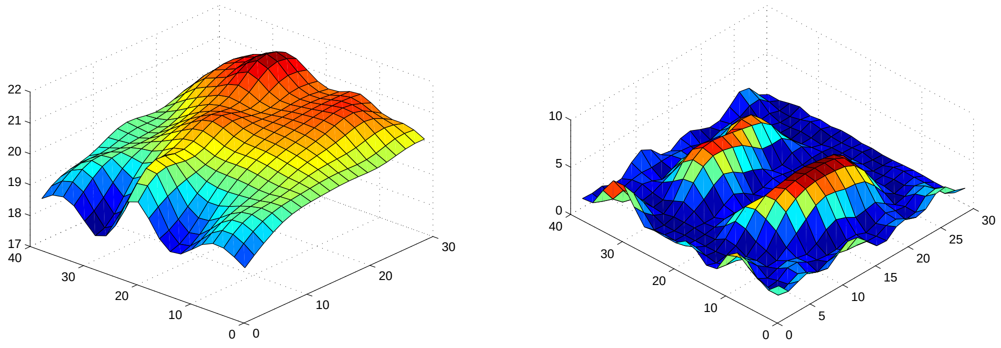
(c) *Precipitation correlations*

Figure 1: (a) A deployment of a sensor network with 54 nodes at the Intel Berkeley Lab. Correlations are often nonstationary as illustrated by (b) temperature data from the sensor network deployment in Figure 1(a), showing the correlation between a sensor placed on the blue square and other possible locations; (c) precipitation data from measurements made across the Pacific Northwest, Figure 11(b).

The paper is organized as follows. In Section 2, we introduce Gaussian Processes. We review mutual information criterion in Section 3, and describe our approximation algorithm to optimize mutual information in Section 4. Section 5 presents several approaches towards making the optimization more computationally efficient. In Section 6, we discuss how we can extend mutual information to be robust against node failures and uncertainty in the model. Section 8 relates our approach to other possible optimization criteria, and Section 7 describes related work. Section 9 presents our experiments.

2. Gaussian Processes

In this section, we review *Gaussian Processes*, the probabilistic model for spatial phenomena that forms the basis of our sensor placement algorithms.



(a) *Temperature prediction using GP* (b) *Variance of temperature prediction*

Figure 2: Posterior mean and variance of the temperature GP estimated using all sensors: (a) Predicted temperature; (b) predicted variance.

2.1 Modeling Sensor Data Using the Multivariate Normal Distribution

Consider, for example, the sensor network we deployed as shown in Figure 1(a) that measures a temperature field at 54 discrete locations. In order to predict the temperature at one of these locations from the other sensor readings, we need the joint distribution over temperatures at the 54 locations. A simple, yet often effective (cf. Deshpande et al., 2004), approach is to assume that the temperatures have a (multivariate) Gaussian joint distribution. Denoting the set of locations as $\mathcal{V}$, in our sensor network example $|\mathcal{V}| = 54$, we have a set of $n = |\mathcal{V}|$ corresponding random variables $\mathcal{X}_{\mathcal{V}}$ with joint distribution:

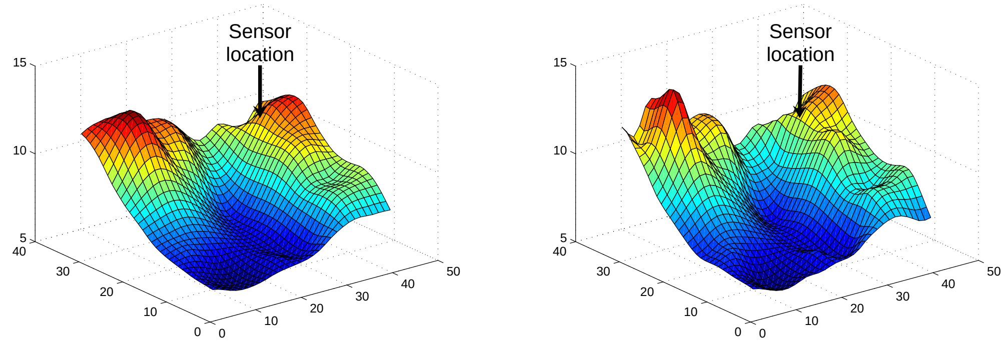
$$P(\mathcal{X}_{\mathcal{V}} = \mathbf{x}_{\mathcal{V}}) = \frac{1}{(2\pi)^{n/2}|\Sigma_{\mathcal{V}\mathcal{V}}|} e^{-\frac{1}{2}(\mathbf{x}_{\mathcal{V}}-\mu_{\mathcal{V}})^T \Sigma_{\mathcal{V}\mathcal{V}}^{-1}(\mathbf{x}_{\mathcal{V}}-\mu_{\mathcal{V}})},$$

where $\mu_{\mathcal{V}}$ is the mean vector and $\Sigma_{\mathcal{V}\mathcal{V}}$ is the covariance matrix. Interestingly, if we consider a subset, $\mathcal{A} \subseteq \mathcal{V}$, of our random variables, denoted by $\mathcal{X}_{\mathcal{A}}$, then their joint distribution is also Gaussian.

2.2 Modeling Sensor Data Using Gaussian Processes

In our sensor network example, we are not just interested in temperatures at sensed locations, but also at locations where no sensors were placed. In such cases, we can use regression techniques to perform prediction (Golub and Van Loan, 1989; Hastie et al., 2003). Although linear regression often gives excellent predictions, there is usually no notion of uncertainty about these predictions, for example, for Figure 1(a), we are likely to have better temperature estimates at points near existing sensors, than in the two central areas that were not instrumented. A *Gaussian process* (GP) is a natural generalization of linear regression that allows us to consider uncertainty about predictions.

Intuitively, a GP generalizes multivariate Gaussians to an infinite number of random variables. In analogy to the multivariate Gaussian above where the index set $\mathcal{V}$ was finite, we now have a (possibly uncountably) infinite index set $\mathcal{V}$. In our temperature example, $\mathcal{V}$ would be a subset of $\mathbb{R}^2$, and



(a) *Example kernel function.* (b) *Data from the empirical covariance matrix.*

Figure 3: Example kernel function learned from the Berkeley Lab temperature data: (a) learned covariance function $\mathcal{K}(x,\cdot)$, where x is the location of sensor 41; (b) "ground truth", interpolated empirical covariance values for the same sensors. Observe the close match between predicted and measured covariances.

each index would correspond to a position in the lab. GPs have been widely studied (cf. MacKay, 2003; Paciorek, 2003; Seeger, 2004; O'Hagan, 1978; Shewry and Wynn, 1987; Lindley and Smith, 1972), and generalize Kriging estimators commonly used in geostatistics (Cressie, 1991).

An important property of GPs is that for every finite subset $\mathcal{A}$ of the indices $\mathcal{V}$, which we can think about as locations in the plane, the joint distribution over the corresponding random variables $\mathcal{X}_\mathcal{A}$ is Gaussian, for example, the joint distribution over temperatures at a finite number of sensor locations is Gaussian. In order to specify this distribution, a GP is associated with a *mean function* $\mathcal{M}(\cdot)$, and a symmetric positive-definite *kernel function* $\mathcal{K}(\cdot,\cdot)$, often called the covariance function. For each random variable with index $u \in \mathcal{V}$, its mean μ_u is given by $\mathcal{M}(u)$. Analogously, for each pair of indices $u, v \in \mathcal{V}$, their covariance σ_{uv} is given by $\mathcal{K}(u,v)$. For simplicity of notation, we denote the mean vector of some set of variables $\mathcal{X}_\mathcal{A}$ by $\mu_\mathcal{A}$, where the entry for element u of $\mu_\mathcal{A}$ is $\mathcal{M}(u)$. Similarly, we denote their covariance matrix by $\Sigma_{\mathcal{A}\mathcal{A}}$, where the entry for u, v is $\mathcal{K}(u,v)$.

The GP representation is extremely powerful. For example, if we observe a set of sensor measurements $\mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}$ corresponding to the finite subset $\mathcal{A} \subset \mathcal{V}$, we can predict the value at any point $y \in \mathcal{V}$ conditioned on these measurements, $P(\mathcal{X}_y \mid x_\mathcal{A})$. The distribution of $\mathcal{X}_y$ given these observations is a Gaussian whose conditional mean $\mu_{y|\mathcal{A}}$ and variance $\sigma^2_{y|\mathcal{A}}$ are given by:

Very important poiint to understand the link between the GP approx and the formulation of the condition entropy

$$\mu_{y|\mathcal{A}} = \mu_y + \Sigma_{y\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}(x_\mathcal{A} - \mu_\mathcal{A}), \tag{1}$$

$$\sigma^2_{y|\mathcal{A}} = \mathcal{K}(y,y) - \Sigma_{y\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\Sigma_{\mathcal{A}y}, \tag{2}$$

where $\Sigma_{y\mathcal{A}}$ is a covariance vector with one entry for each $u \in \mathcal{A}$ with value $\mathcal{K}(y,u)$, and $\Sigma_{\mathcal{A}y} = \Sigma^T_{y\mathcal{A}}$. Figure 2(a) and Figure 2(b) show the posterior mean and variance derived using these equations on 54 sensors at Intel Labs Berkeley. Note that two areas in the center of the lab were not instrumented. These areas have higher posterior variance, as expected. An important property of GPs is that the posterior variance (2) does not depend on the actual observed values $x_\mathcal{A}$. Thus, for a given kernel function, the variances in Figure 2(b) will not depend on the observed temperatures.

2.3 Nonstationarity

In order to compute predictive distributions using (1) and (2), the mean and kernel functions have to be known. The mean function can usually be estimated using regression techniques. Estimating kernel functions is difficult, and usually, strongly limiting assumptions are made. For example, it is commonly assumed that the kernel $\mathcal{K}(u,v)$ is *stationary*, which means that the kernel depends only on the difference between the locations, considered as vectors v, u, that is, $\mathcal{K}(u,v) = \mathcal{K}_\theta(u-v)$. Hereby, θ is a set of parameters. Very often, the kernel is even assumed to be *isotropic*, which means that the covariance only depends on the distance between locations, that is, $\mathcal{K}(u,v) = \mathcal{K}_\theta(||u-v||_2)$. Common choices for isotropic kernels are the exponential kernel, $\mathcal{K}_\theta(\delta) = \exp(-\frac{|\delta|}{\theta})$, and the Gaussian kernel, $\mathcal{K}_\theta(\delta) = \exp(-\frac{\delta^2}{\theta^2})$. These assumptions are frequently strongly violated in practice, as illustrated in the real sensor data shown in Figures 1(b) and 1(c). In Section 8.1, we discuss how placements optimized from models with isotropic kernels reduce to geometric covering and packing problems.

In this paper, we *do not* assume that $\mathcal{K}(\cdot,\cdot)$ is stationary or isotropic. Our approach is general, and can use *any* kernel function. In our experiments, we use the approach of Nott and Dunsmuir (2002) to estimate nonstationary kernels from data collected by an initial deployment. More specifically, their assumption is that an estimate of the empirical covariance $\Sigma_{\mathcal{A}\mathcal{A}}$ at a set of observed locations is available, and that the process can be locally described by a collection of isotropic processes, associated with a set of reference points. An example of a kernel function estimated using this method is presented in Figure 3(a). In Section 9.2, we show that placements based on such nonstationary GPs lead to far better prediction accuracies than those obtained from isotropic kernels.

3. Optimizing Sensor Placements

Usually, we are limited to deploying a small number of sensors, and thus must carefully choose where to place them. In spatial statistics this optimization is called *sampling* or *experimental design*: finding the k best sensor locations out of a finite subset $\mathcal{V}$ of possible locations, for example, out of a grid discretization of $\mathbb{R}^2$.

3.1 The Entropy Criterion

We first have to define what a good design is. Intuitively, we want to place sensors which are most informative with respect to the entire design space. A natural notion of uncertainty is the conditional entropy of the unobserved locations $\mathcal{V}\setminus\mathcal{A}$ after placing sensors at locations $\mathcal{A}$,

$$H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_\mathcal{A}) = -\int p(\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}}, \mathbf{x}_\mathcal{A}) \log p(\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathbf{x}_\mathcal{A}) d\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}} d\mathbf{x}_\mathcal{A}, \tag{3}$$

where we use $\mathcal{X}_\mathcal{A}$ and $\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}$ to refer to sets of random variables at the locations $\mathcal{A}$ and $\mathcal{V}\setminus\mathcal{A}$. Intuitively, minimizing this quantity aims at finding the placement which results in the lowest uncertainty about all uninstrumented locations $\mathcal{V}\setminus\mathcal{A}$ after observing the placed sensors $\mathcal{A}$. A good placement would therefore minimize this conditional entropy, that is, we want to find

$$\mathcal{A}^* = \text{argmin}_{\mathcal{A}\subset\mathcal{V}:|\mathcal{A}|=k} H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_\mathcal{A}).$$

Using the identity $H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}) = H(\mathcal{X}_{\mathcal{V}}) - H(\mathcal{X}_{\mathcal{A}})$, we can see that

$$\mathcal{A}^* = \operatorname{argmin}_{\mathcal{A}\subset\mathcal{V}:|\mathcal{A}|=k} H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}) = \operatorname{argmax}_{\mathcal{A}\subset\mathcal{V}:|\mathcal{A}|=k} H(\mathcal{X}_{\mathcal{A}}).$$

So we can see that we need to find a set of sensors $\mathcal{A}$ which is most uncertain about each other. Unfortunately, this optimization problem, often also referred to as D-optimal design in the experiment design literature (cf. Currin et al., 1991), has been shown to be NP-hard (Ko et al., 1995):

Theorem 1 (Ko et al., 1995) *Given rational M and rational covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$ over Gaussian random variables $\mathcal{V}$, deciding whether there exists a subset $\mathcal{A} \subseteq \mathcal{V}$ of cardinality k such that $H(\mathcal{X}_{\mathcal{A}}) \geq M$ is NP-complete.*

Therefore, the following greedy heuristic has found common use (McKay et al., 1979; Cressie, 1991): One starts from an empty set of locations, $\mathcal{A}_0 = \emptyset$, and greedily adds placements until $|\mathcal{A}| = k$. At each iteration, starting with set $\mathcal{A}_i$, the greedy rule used is to add the location $y_H^* \in \mathcal{V} \setminus \mathcal{A}$ that has highest conditional entropy,

$$y_H^* = \operatorname{argmax}_y H(\mathcal{X}_y \mid \mathcal{X}_{\mathcal{A}_i}), \tag{4}$$

that is, the location we are most uncertain about given the sensors placed thus far. If the set of selected locations at iteration i is $\mathcal{A}_i = \{y_1, \ldots, y_i\}$, using the chain-rule of entropies, we have that:

$$H(\mathcal{X}_{\mathcal{A}_i}) = H(\mathcal{X}_{y_i} \mid \mathcal{X}_{\mathcal{A}_{i-1}}) + \ldots + H(\mathcal{X}_{y_2} \mid \mathcal{X}_{\mathcal{A}_1}) + H(\mathcal{X}_{y_1} \mid \mathcal{X}_{\mathcal{A}_0}).$$

Note that the (differential) entropy of a Gaussian random variable $\mathcal{X}_y$ conditioned on some set of variables $\mathcal{X}_{\mathcal{A}}$ is a monotonic function of its variance:

$$H(\mathcal{X}_y \mid \mathcal{X}_{\mathcal{A}}) = \frac{1}{2}\log(2\pi e \sigma^2_{\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}}) = \frac{1}{2}\log\sigma^2_{\mathcal{X}_y|\mathcal{X}_{\mathcal{A}}} + \frac{1}{2}(\log(2\pi)+1), \tag{5}$$

which can be computed in closed form using Equation (2). Since for a fixed kernel function, the variance does not depend on the observed values, this optimization can be done before deploying the sensors, that is, a sequential, closed-loop design taking into account previous measurements bears no advantages over an open-loop design, performed before any measurements are made.

3.2 An Improved Design Criterion: Mutual Information

The entropy criterion described above is intuitive for finding sensor placements, since the sensors that are most uncertain about each other should cover the space well. Unfortunately, this entropy criterion suffers from the problem shown in Figure 4, where sensors are placed far apart along the boundary of the space. Since we expect predictions made from a sensor measurement to be most precise in a region around it, such placements on the boundary are likely to “waste” information. This phenomenon has been noticed previously by Ramakrishnan et al. (2005), who proposed a weighting heuristic. Intuitively, this problem arises because the entropy criterion is *indirect*: the criterion only considers the entropy of the selected sensor locations, rather than considering prediction quality over the space of interest. This indirect quality of the entropy criterion is surprising, since

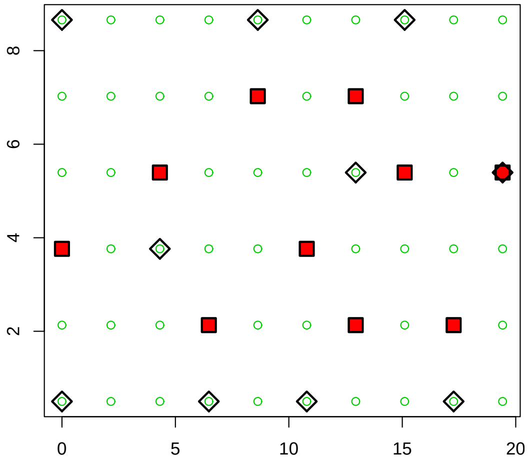


Figure 4: An example of placements chosen using entropy and mutual information criteria on a subset of the temperature data from the Intel deployment. Diamonds indicate the positions chosen using entropy; squares the positions chosen using MI.

the criterion was derived from the "predictive" formulation $H(\mathcal{V}\setminus\mathcal{A} \mid \mathcal{A})$ in Equation (3), which is equivalent to maximizing $H(\mathcal{A})$.

Caselton and Zidek (1984) proposed a different optimization criterion, which searches for the subset of sensor locations that most significantly reduces the uncertainty about the estimates in the rest of the space. More formally, we consider our space as a discrete set of locations $\mathcal{V} = \mathcal{S} \cup \mathcal{U}$ composed of two parts: a set $\mathcal{S}$ of possible positions where we can place sensors, and another set $\mathcal{U}$ of positions of interest, where no sensor placements are possible. The goal is to place a set of k sensors that will give us good predictions at all uninstrumented locations $\mathcal{V}\setminus\mathcal{A}$. Specifically, we want to find

$$\mathcal{A}^* = \text{argmax}_{\mathcal{A}\subseteq\mathcal{S}:|\mathcal{A}|=k} H(X_{\mathcal{V}\setminus\mathcal{A}}) - H(X_{\mathcal{V}\setminus\mathcal{A}} \mid X_{\mathcal{A}}),$$

that is, the set $\mathcal{A}^*$ that maximally reduces the entropy over the rest of the space $\mathcal{V}\setminus\mathcal{A}^*$. Note that this criterion $H(X_{\mathcal{V}\setminus\mathcal{A}}) - H(X_{\mathcal{V}\setminus\mathcal{A}} \mid X_{\mathcal{A}})$ is equivalent to finding the set that maximizes the *mutual information* $I(X_{\mathcal{A}}; X_{\mathcal{V}\setminus\mathcal{A}})$ between the locations $\mathcal{A}$ and the rest of the space $\mathcal{V}\setminus\mathcal{A}$. In their follow-up work, Caselton et al. (1992) and Zidek et al. (2000), argue against the use of mutual information in a setting where the entropy $H(X_{\mathcal{A}})$ in the observed locations constitutes a significant part of the total uncertainty $H(X_{\mathcal{V}})$. Caselton et al. (1992) also argue that, in order to compute $\text{MI}(\mathcal{A})$, one needs an accurate model of $P(X_{\mathcal{V}})$. Since then, the entropy criterion has been dominantly used as a placement criterion. Nowadays however, the estimation of complex nonstationary models for $P(X_{\mathcal{V}})$, as well as computational aspects, are very well understood and handled. Furthermore, we show empirically, that even in the sensor selection case, mutual information outperforms entropy on several practical placement problems.

On the same simple example in Figure 4, this mutual information criterion leads to intuitively appropriate central sensor placements that do not have the "wasted information" property of the entropy criterion. Our experimental results in Section 9 further demonstrate the advantages in performance of the mutual information criterion. For simplicity of notation, we will often use $\text{MI}(\mathcal{A}) = I(X_{\mathcal{A}}; X_{\mathcal{V}\setminus\mathcal{A}})$ to denote the mutual information objective function. Notice that in this no-

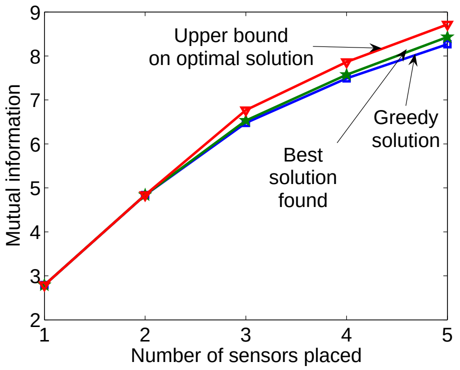


Figure 5: Comparison of the greedy algorithm with the optimal solutions on a small problem. We select from 1 to 5 sensor locations out of 16, on the Intel Berkeley temperature data set as discussed in Section 9. The greedy algorithm is always within 95 percent of the optimal solution.

tation the process $\mathcal{X}$ and the set of locations $\mathcal{V}$ is implicit. We will also write $H(\mathcal{A})$ instead of $H(\mathcal{X}_{\mathcal{A}})$.

The mutual information is also hard to optimize:

Theorem 2 *Given rational M and a rational covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$ over Gaussian random variables $\mathcal{V} = \mathcal{S} \cup \mathcal{U}$, deciding whether there exists a subset $\mathcal{A} \subseteq \mathcal{S}$ of cardinality k such that $\mathrm{MI}(\mathcal{A}) \geq M$ is NP-complete.*

Proofs of all results are given in Appendix A. Due to the problem complexity, we cannot expect to find optimal solutions in polynomial time. However, if we implement the simple greedy algorithm for the mutual information criterion (details given below), and optimize designs on real-world placement problems, we see that the greedy algorithm gives almost optimal solutions, as presented in Figure 5. In this small example, where we could compute the optimal solution, the performance of the greedy algorithm was at most five percent worse than the optimal solution. In the following sections, we will give theoretical bounds and empirical evidence justifying this near-optimal behavior.

4. Approximation Algorithm

Optimizing the mutual information criterion is an NP-complete problem. We now describe a polynomial time algorithm with a constant-factor approximation guarantee.

4.1 The Algorithm

Our algorithm is greedy, simply adding sensors in sequence, choosing the next sensor which provides the maximum increase in mutual information. More formally, using $\mathrm{MI}(\mathcal{A}) = I(\mathcal{X}_{\mathcal{A}}; \mathcal{X}_{\mathcal{V}\setminus\mathcal{A}})$,

```
Input: Covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, $k$, $\mathcal{V} = \mathcal{S} \cup \mathcal{U}$
Output: Sensor selection $\mathcal{A} \subseteq \mathcal{S}$
begin
    $\mathcal{A} \leftarrow \emptyset$;
    for $j$ = 1 to $k$ do
        for $y \in \mathcal{S} \setminus \mathcal{A}$ do $\delta_y \leftarrow \dfrac{\sigma_y^2 - \Sigma_{y\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}\Sigma_{\mathcal{A}y}}{\sigma_y^2 - \Sigma_{y\bar{\mathcal{A}}}\Sigma_{\bar{\mathcal{A}}\bar{\mathcal{A}}}^{-1}\Sigma_{\bar{\mathcal{A}}y}}$;
        $y^* \leftarrow \mathrm{argmax}_{y \in \mathcal{S} \setminus \mathcal{A}}\, \delta_y$;
        $\mathcal{A} \leftarrow \mathcal{A} \cup y^*$;
end
```

Algorithm 1: Approximation algorithm for maximizing mutual information.

our goal is to greedily select the next sensor y that maximizes:

$$\begin{aligned}\mathrm{MI}(\mathcal{A}\cup y) - \mathrm{MI}(\mathcal{A}) &= H(\mathcal{A}\cup y) - H(\mathcal{A}\cup y \mid \bar{\mathcal{A}}) - \left[H(\mathcal{A}) - H(\mathcal{A} \mid \bar{\mathcal{A}}\cup y)\right]\\ &= H(\mathcal{A}\cup y) - H(\mathcal{V}) + H(\bar{\mathcal{A}}) - \left[H(\mathcal{A}) - H(\mathcal{V}) + H(\bar{\mathcal{A}}\cup y)\right]\\ &= H(y \mid \mathcal{A}) - H(y \mid \bar{\mathcal{A}}), \qquad (6)\end{aligned}$$

where, to simplify notation, we write $\mathcal{A}\cup y$ to denote the set $\mathcal{A}\cup\{y\}$, and use $\bar{\mathcal{A}}$ to mean $\mathcal{V}\setminus(\mathcal{A}\cup y)$. Note that the greedy rule for entropy in Equation (4) only considers the $H(y \mid \mathcal{A})$ part of Equation (6), measuring the uncertainty of location y with respect to the placements $\mathcal{A}$. In contrast, the greedy mutual information trades off this uncertainty with $-H(y \mid \bar{\mathcal{A}})$, which forces us to pick a y that is "central" with respect to the unselected locations $\bar{\mathcal{A}}$, since those "central" locations will result in the least conditional entropy $H(y \mid \bar{\mathcal{A}})$. Using the definition of conditional entropy in Equation (5), Algorithm 1 shows our greedy sensor placement algorithm.

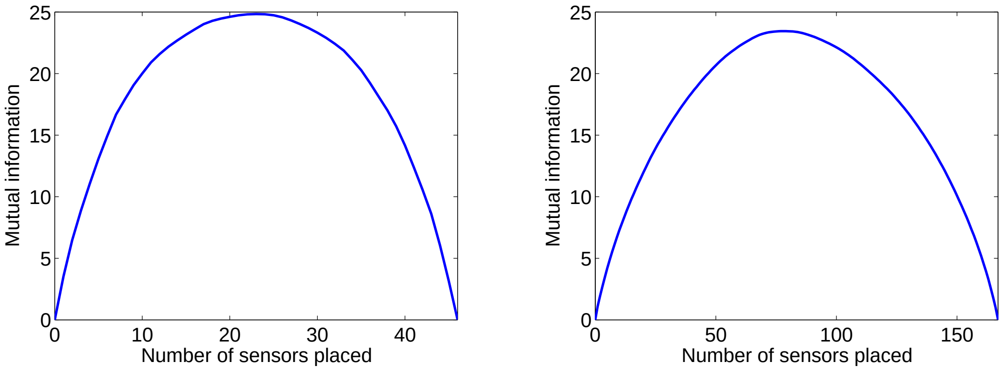
4.2 An Approximation Bound

We now prove that, if the discretization $\mathcal{V}$ of locations of interest in the Gaussian process is fine enough, our greedy algorithm gives a $(1-1/e)$ approximation, approximately 63% of the optimal sensor placement: If the algorithm returns set $\widehat{A}$, then

$$\mathrm{MI}(\widehat{A}) \geq (1-1/e) \max_{\mathcal{A}\subset\mathcal{S},|\mathcal{A}|=k} \mathrm{MI}(\mathcal{A}) - k\varepsilon,$$

for some small $\varepsilon > 0$. To prove this result, we use *submodularity* (cf. Nemhauser et al., 1978). Formally, a set function F is called *submodular*, if for all $\mathcal{A},\mathcal{B} \subseteq \mathcal{V}$ it holds that $F(\mathcal{A}\cup\mathcal{B}) + F(\mathcal{A}\cap\mathcal{B}) \leq F(\mathcal{A}) + F(\mathcal{B})$. Equivalently, using an induction argument as done by Nemhauser et al. (1978), a set function is submodular if for all $\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{V}$ and $y \in \mathcal{V}\setminus\mathcal{A}'$ it holds that $F(\mathcal{A}\cup y) - F(\mathcal{A}) \geq F(\mathcal{A}'\cup y) - F(\mathcal{A}')$. This second characterization intuitively represents "diminishing returns": adding a sensor y when we only have a small set of sensors $\mathcal{A}$ gives us more advantage than adding y to a larger set of sensors $\mathcal{A}'$. Using the "information never hurts" bound, $H(y \mid \mathcal{A}) \geq H(y \mid \mathcal{A}\cup\mathcal{B})$ (Cover and Thomas, 1991), note that our greedy update rule maximizing $H(y \mid \mathcal{A}) - H(y \mid \bar{\mathcal{A}})$ implies

$$\mathrm{MI}(\mathcal{A}'\cup y) - \mathrm{MI}(\mathcal{A}') \leq \mathrm{MI}(\mathcal{A}\cup y) - \mathrm{MI}(\mathcal{A}),$$



(a) *Temperature data* (b) *Precipitation data*

Figure 6: Mutual information of greedy sets of increasing size. It can be seen that clearly mutual information is not monotonic. MI is monotonic, however, in the initial part of the curve corresponding to small placements. This allows us to prove approximate monotonicity.

whenever $\mathcal{A} \subseteq \mathcal{A}'$, and any $y \in \mathcal{V} \setminus \mathcal{A}'$, that is, adding y to $\mathcal{A}$ helps more than adding y to $\mathcal{A}'$. In fact, this inequality holds for arbitrary sets $\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{V}$ and $y \in \mathcal{V} \setminus \mathcal{A}'$, not just for the sets considered by the greedy algorithm. Hence we have shown:

Lemma 3 *The set function $\mathcal{A} \mapsto \mathrm{MI}(\mathcal{A})$ is submodular.*

A submodular set function F is called monotonic if $F(\mathcal{A} \cup y) \geq F(\mathcal{A})$ for $y \in \mathcal{V}$. For such functions, Nemhauser et al. (1978) prove the following fundamental result:

Theorem 4 (Nemhauser et al., 1978) *Let F be a monotone submodular set function over a finite ground set $\mathcal{V}$ with $F(\emptyset) = 0$. Let $\mathcal{A}_G$ be the set of the first k elements chosen by the greedy algorithm, and let $\mathrm{OPT} = \max_{\mathcal{A} \subset \mathcal{V}, |\mathcal{A}| = k} F(\mathcal{A})$. Then*

$$F(\mathcal{A}_G) \geq \left(1 - \left(\frac{k-1}{k}\right)^k\right) \mathrm{OPT} \geq (1 - 1/e)\,\mathrm{OPT}.$$

Hence the greedy algorithm guarantees a performance guarantee of $(1 - 1/e)\,\mathrm{OPT}$, where OPT is the value of the optimal subset of size k. This greedy algorithm is defined by selecting in each step the element $y^* = \mathrm{argmax}_y F(\mathcal{A} \cup y) - F(\mathcal{A})$. This is exactly the algorithm we proposed in the previous section for optimizing sensor placements (Algorithm 1).

Clearly, $\mathrm{MI}(\emptyset) = I(\emptyset; \mathcal{V}) = 0$, as required by Theorem 4. However, the monotonicity of mutual information is not apparent. Since $\mathrm{MI}(\mathcal{V}) = I(\mathcal{V}, \emptyset) = 0$, the objective function will increase and then decrease, and, thus, is not monotonic, as shown in Figures 6(a) and 6(b). Fortunately, the proof of Nemhauser et al. (1978) does not use monotonicity for all possible sets, it is sufficient to prove that MI is monotonic for all sets of size up to $2k$. Intuitively, mutual information is not monotonic when the set of sensor locations approaches $\mathcal{V}$. If the discretization level is significantly larger than $2k$ points, then mutual information should meet the conditions of the proof of Theorem 4.

Thus the heart of our analysis of Algorithm 1 will be to prove that if the discretization of the Gaussian process is fine enough, then mutual information is *approximately monotonic* for sets of size up to $2k$. More precisely, we prove the following result:

Lemma 5 *Let $\mathcal{X}$ be a Gaussian process on a compact subset $\mathcal{C}$ of $\mathbb{R}^m$ with a positive-definite, continuous covariance kernel $\mathcal{K} : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}_0^+$. Assume the sensors have a measurement error with variance at least σ^2. Then, for any $\varepsilon > 0$, and any finite maximum number k of sensors to place there exists a discretization $\mathcal{V} = \mathcal{S} \cup \mathcal{U}$, $\mathcal{S}$ and $\mathcal{U}$ having mesh width δ such that $\forall y \in \mathcal{V} \setminus \mathcal{A}, \mathrm{MI}(\mathcal{A} \cup y) \geq \mathrm{MI}(\mathcal{A}) - \varepsilon$ for all $\mathcal{A} \subseteq \mathcal{S}$, $|\mathcal{A}| \leq 2k$.*

If the covariance function is Lipschitz-continuous, such as the Gaussian Radial Basis Function (RBF) kernel, the following corollary gives a bound on the required discretization level with respect to the Lipschitz constant:

Corollary 6 *If $\mathcal{K}$ is Lipschitz-continuous with constant L, then the required discretization is*

$$\delta \leq \frac{\varepsilon\sigma^6}{4kLM\left(\sigma^2 + 2k^2M + 6k^2\sigma^2\right)},$$

where $M = \max_{x \in C} \mathcal{K}(x, x)$, for $\varepsilon < \min(M, 1)$.

Corollary 6 guarantees that for any $\varepsilon > 0$, a polynomial discretization level is sufficient to guarantee that mutual information is $\varepsilon-$approximately monotonic. These bounds on the discretization are, of course, worst case bounds. The worst-case setting occurs when the sensor placements $\mathcal{A}$ are arbitrarily close to each other, since the entropy part $H(y \mid \mathcal{A})$ in Equation (6) can become negative. Since most GPs are used for modeling physical phenomena, both the optimal sensor placement and the sensor placement produced by the greedy algorithm can be expected to be spread out, and not condensed to a small region of the sensing area. Hence we expect the bounds to be very loose in the situations that arise during normal operation of the greedy algorithm.

Combining our Lemmas 3 and 5 with Theorem 4, we obtain our constant-factor approximation bound on the quality of the sensor placements obtained by our algorithm:

Theorem 7 *Under the assumptions of Lemma 5, Algorithm 1 is guaranteed to select a set $\mathcal{A}$ of k sensors for which*

$$\mathrm{MI}(\mathcal{A}) \geq (1 - 1/e)(\mathrm{OPT} - k\varepsilon),$$

where OPT *is the value of the mutual information for the optimal placement.*

Note that our bound has two implications: First, it shows that our greedy algorithm has a guaranteed minimum performance level of $1 - 1/e$ when compared to the optimal solution. Second, our approach also provides an upper-bound on the value of the optimal placement, which can be used to bound the quality of the placements by other heuristic approaches, such as local search, that may perform better than our greedy algorithm on specific problems.

4.3 Sensor Placement with Non-constant Cost Functions

In many real-world settings, the cost of placing a sensor depends on the specific location. Such cases can often be formalized by specifying a total budget L, and the task is to select placements $\mathcal{A}$ whose total cost $c(\mathcal{A})$ is within our budget. Recently, the submodular function maximization approach of Nemhauser et al. (1978) has been extended to address this budgeted case (Sviridenko, 2004; Krause and Guestrin, 2005), in the case of modular cost functions, that is, $c(\mathcal{A}) = \sum_{i=1}^{k} c(\mathcal{X}_i)$, where $\mathcal{A} = \{\mathcal{X}_1, \ldots, \mathcal{X}_k\}$ and $c(\mathcal{X}_i)$ is the cost for selecting element $\mathcal{X}_i$. The combination of the analysis in this paper with these new results also yields a constant-factor $(1-1/e)$ approximation guarantee for the sensor placement problem with non-uniform costs.

The algorithm for this budgeted case first enumerates all subsets of cardinality at most three. For each of these candidate subsets, we run a greedy algorithm, which adds elements until the budget is exhausted. The greedy rule optimizes a benefit cost ratio, picking the element for which the increase of mutual information divided by the cost of placing the sensor is maximized: More formally, at each step, the greedy algorithm adds the element y^* such that

$$y^* = \operatorname{argmax}_{y \in \mathcal{S} \setminus \mathcal{A}} \frac{H(y \mid \mathcal{A}) - H(y \mid \bar{\mathcal{A}})}{c(y)}.$$

Krause and Guestrin (2005) show that this algorithm achieves an approximation guarantee of

$$(1-1/e)\,\mathrm{OPT} - \frac{2L\varepsilon}{c_{\min}},$$

where L is the available budget, and $c_{\min}$ is the minimum cost of all locations. A requirement for this result to hold is that mutual information is ε-monotonic up to sets of size $\frac{2L}{c_{\min}}$. The necessary discretization level can be established similarly as in Corollary 6, with k replaced by $\frac{L}{c_{\min}}$.

4.4 Online Bounds

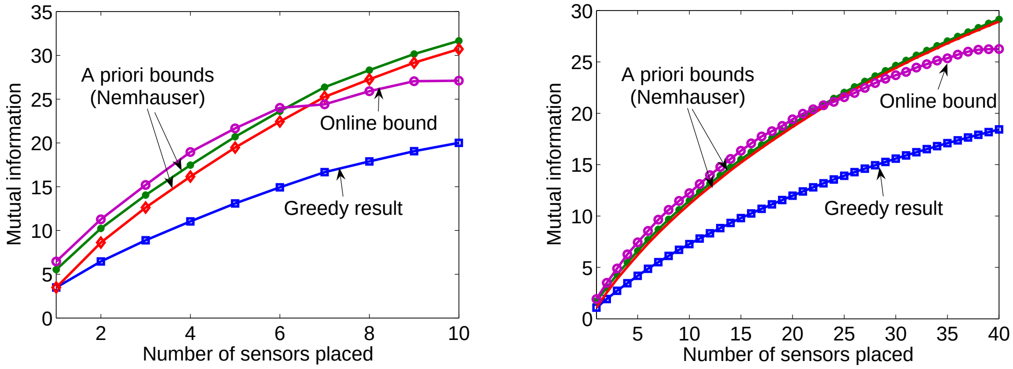
Since mutual information is approximately monotonic and submodular, Theorem 7 proves an *a priori* approximation guarantee of $(1-1/e)$. For most practical problems however, this bound is very loose. The following observation allows to compute online bounds on the optimal value:

Proposition 8 *Assume that the discretization is fine enough to guarantee ε-monotonicity for mutual information, and that the greedy algorithm returns an approximate solution $\mathcal{A}_k$, $|\mathcal{A}_k| = k$. For all $y \in \mathcal{S}$, let $\delta_y = \mathrm{MI}(\mathcal{A} \cup y) - \mathrm{MI}(\mathcal{A})$. Sort the δ_y in decreasing order, and consider the sequence $\delta^{(1)}, \ldots, \delta^{(k)}$ of the first k elements. Then $\mathrm{OPT} \leq \mathrm{MI}(\mathcal{A}_k) + \sum_{i=1}^{k} \delta^{(i)} + k\varepsilon$.*

The proof of this proposition follows directly from submodularity and ε-monotonicity. In many applications, especially for large placements, this bound can be much tighter than the bound guaranteed by Theorem 7. Figures 7(a) and 7(b) compare the a priori and online bounds for the data sets discussed in Section 9.1.

4.5 Exact Optimization and Tighter Bounds Using Mixed Integer Programming

There is another way to get even tighter bounds, or even compute the optimal solution. This approach is based on branch & bound algorithm for solving a mixed integer program for monotonic submodular functions (Nemhauser and Wolsey, 1981). We used this algorithm to bound the value of the optimal solution in Figure 5.



(a) *Temperature data*

(b) *Precipitation data*

Figure 7: Online bounds: mutual information achieved by the greedy algorithm, the $(1-1/e)$ and $1-(1-1/k)^k$ a priori bounds and the online bound described in Section 4.4.

The mixed integer program is given by:

$$\begin{aligned} &\max \eta; \\ \eta &\leq \mathrm{MI}(\mathcal{B}) + \sum_{y_i \in \mathcal{S}\setminus\mathcal{B}} \alpha_i[\mathrm{MI}(\mathcal{B}\cup y_i) - \mathrm{MI}(\mathcal{B})], \quad \forall \mathcal{B} \subseteq \mathcal{S}; && (7) \\ \sum_i \alpha_i &\leq k, \quad \forall i; && (8) \\ \alpha_i &\in \{0,1\}, \quad \forall i; \end{aligned}$$

where $\alpha_i = 1$ means that location y_i should be selected. Note that this MIP can be easily extended to handle the case in which each location can have a different cost, by replacing the constraint (8) by $\sum_i \alpha_i c_i \leq L$, where L is the budget and $c_i = c(y_i)$.

Unfortunately, this MIP has exponentially many constraints. Nemhauser and Wolsey (1981) proposed the following constraint generation algorithm: Let $\alpha^{\mathcal{A}}$ denote an assignment to $\alpha_1, \ldots, \alpha_n$ such that $\alpha_i = 1$ iff $y_i \in \mathcal{A}$. Starting with no constraints of type (7), the MIP is solved, and one checks whether the current solution $(\eta, \alpha^{\mathcal{B}})$ satisfies $\eta \leq \mathrm{MI}(\mathcal{B})$. If it does not, a violated constraint has been found. Since solving individual instances (even with only polynomially many constraints) is NP-hard, we need to resort to search heuristics such as Branch and Bound and Cut during the constraint generation process.

The analysis of this MIP, as presented by Nemhauser and Wolsey (1981), assumes monotonicity. In the case of mutual information, the objective is only approximately monotonic. In particular, consider a a placement defined by $\alpha^{\mathcal{A}}$. Then, by submodularity, for all $\mathcal{B}$, we have that:

$$\begin{aligned} \mathrm{MI}(\mathcal{B}) + \sum_{y_i \in \mathcal{S}\setminus\mathcal{B}} \alpha_i^{\mathcal{A}}[\mathrm{MI}(\mathcal{B}\cup y_i) - \mathrm{MI}(\mathcal{B})] &= \mathrm{MI}(\mathcal{B}) + \sum_{y_i \in \mathcal{A}\setminus\mathcal{B}} [\mathrm{MI}(\mathcal{B}\cup y_i) - \mathrm{MI}(\mathcal{B})], \\ &\geq \mathrm{MI}(\mathcal{A}\cup\mathcal{B}). \end{aligned}$$

By approximate monotonicity:

$$\mathrm{MI}(\mathcal{A}\cup\mathcal{B}) \geq \mathrm{MI}(\mathcal{A}) - k\varepsilon.$$

Thus, $(\hat{\eta}, \alpha^{\mathcal{A}})$, for $\hat{\eta} \leq \mathrm{MI}(\mathcal{A}) - k\varepsilon$ is a feasible solution for the mixed integer program. Since we are maximizing η, for the optimal solution $(\eta^*, \alpha^{\mathcal{A}^*})$ of the MIP it holds that

$$\mathrm{MI}(\mathcal{A}^*) \geq \mathrm{OPT} - k\varepsilon.$$

There is another MIP formulation for maximizing general submodular functions without the ε-monotonicity requirement. The details can be found in Nemhauser and Wolsey (1981). We however found this formulation to produce much looser bounds, and to take much longer to converge.

5. Scaling Up

Greedy updates for both entropy and mutual information require the computation of conditional entropies using Equation (5), which involves solving a system of $|\mathcal{A}|$ linear equations. For entropy maximization, where we consider $H(y \mid \mathcal{A})$ alone, the complexity of this operation is $O(k^3)$. To maximize the mutual information, we also need $H(y \mid \bar{\mathcal{A}})$ requiring $O(n^3)$, for $n = |\mathcal{V}|$. Since we need to recompute the score of all possible locations at every iteration of Algorithm 1, the complexity of our greedy approach for selecting k sensors is $O(kn^4)$, which is not computationally feasible for very fine discretizations (large n). In Section 5.1 we propose a lazy strategy which often allows to reduce the number of evaluations of the greedy rule, thereby often reducing the complexity to $O(kn^3)$. In Section 5.2 we present a way of exploiting the problem structure by using local kernels, which often reduces the complexity to $O(kn)$. Both approaches can be combined for even more efficient computation.

5.1 Lazy Evaluation Using Priority Queues

It is possible to improve the performance of Algorithm 1 directly under certain conditions by lazy evaluation of the incremental improvements in Line 1. A similar algorithm has been proposed by Robertazzi and Schwartz (1989) in the context of D-optimal design. At the start of the algorithm, all δ_y will be initialized to $+\infty$. The algorithm will maintain information about which δ_y are current, that is, have been computed for the current locations $\mathcal{A}$. Now, the greedy rule in Line 2 will find the node y largest δ_y. If this δ_y has not been updated for the current $\mathcal{A}$, the value is updated and reintroduced into the queue. This process is iterated until the location with maximum δ_y is has an updated value. The algorithm is presented in Algorithm 2. The correctness of this lazy procedure directly follows from submodularity: For a fixed location y, the sequence δ_y must be monotonically decreasing during course of the algorithm.

To understand the efficacy of this procedure, consider the following intuition: If a location y^* is selected, nearby locations will become significantly less desirable and their marginal increases δ_y will decrease significantly. When this happens, these location will not be considered as possible maxima for the greedy step for several iterations. This approach can save significant computation time—we have noticed a decrease of mutual information computations by a factor of six in our experiments described in Section 9.6.

This approach can be efficiently implemented by using a priority queue to maintain the advantages δ_y. Line 2 calls **deletemax** with complexity $O(\log n)$ and Line 3 uses the **insert** operation with com-

```
    Input: Covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, $k$, $\mathcal{V} = \mathcal{S} \cup \mathcal{U}$
    Output: Sensor selection $\mathcal{A} \subseteq \mathcal{S}$
    begin
        $\mathcal{A} \leftarrow \emptyset$;
        foreach $y \in \mathcal{S}$ do $\delta_y \leftarrow +\infty$;
        for $j$ = $1$ to $k$ do
1           foreach $y \in \mathcal{S} \setminus \mathcal{A}$ do $current_y \leftarrow$ false;
            while true do
2               $y^* \leftarrow \mathrm{argmax}_{y \in \mathcal{S} \setminus \mathcal{A}} \delta_y$;
                if $current_{y^*}$ then break;_
3               $\delta_{y^*} \leftarrow H(y \mid \mathcal{A}) - H(y \mid \bar{\mathcal{A}})$ ;
                 $current_{y^*} \leftarrow$ true
            $\mathcal{A} \leftarrow \mathcal{A} \cup y^*$;
    end
```

Algorithm 2: Approximation algorithm for maximizing mutual information efficiently using lazy evaluation.

plexity $O(1)$. Also, as stated Line 1 has an $O(n)$ complexity, and was introduced for simplicity of exposition. In reality, we annotate the δ_y's with the last iteration that they were updated, completely eliminating this step.

5.2 Local Kernels

In this section, we exploit locality in the kernel function to speed up the algorithm significantly: First, we note that, for many GPs, correlation decreases exponentially with the distance between points. Often, variables which are far apart are actually independent. These weak dependencies can be modeled using a covariance function $\mathcal{K}$ for which $\mathcal{K}(x,\cdot)$ has compact support, that is, that has non-zero value only for a small portion of the space. For example, consider the following isotropic covariance function proposed by Storkey (1999):

$$\mathcal{K}(x,y) = \begin{cases} \frac{(2\pi-\Delta)(1+(\cos\Delta)/2)+\frac{3}{2}\sin\Delta}{3\pi}, & \text{for } \Delta < 2\pi, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where $\Delta = \beta \|x-y\|_2$, for $\beta > 0$. This covariance function resembles the Gaussian kernel $\mathcal{K}(x,y) = \exp(-\beta\|x-y\|_2^2/(2\pi))$ as shown in Figure 8, but is zero for distances larger than $2\pi/\beta$.

Even if the covariance function does not have compact support, it can be appropriate to compute $H(y \mid \tilde{\mathcal{B}}) \approx H(y \mid \mathcal{B})$ where $\tilde{\mathcal{B}}$ results from removing all elements x from $\mathcal{B}$ for which $|\mathcal{K}(x,y)| \leq \varepsilon$ for some small value of ε. This truncation is motivated by noting that:

$$\sigma^2_{y|\mathcal{B}\setminus x} - \sigma^2_{y|\mathcal{B}} \leq \frac{\mathcal{K}(y,x)^2}{\sigma^2_x} \leq \frac{\varepsilon^2}{\sigma^2_x}.$$

This implies that the decrease in entropy $H(y \mid \mathcal{B}\setminus x) - H(y \mid \mathcal{B})$ is at most $\varepsilon^2/(\sigma^2\sigma^2_x)$ (using a similar argument as the one in the proof of Lemma 5), assuming that each sensor has independent Gaussian

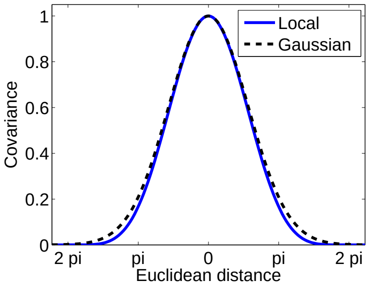


Figure 8: Comparison of local and Gaussian kernels.

```
    **Input**: Covariance $\Sigma_{\mathcal{V}\mathcal{V}}$, $k$, $\mathcal{V} = \mathcal{S} \cup \mathcal{U}, \varepsilon > 0$
    **Output**: Sensor selection $\mathcal{A} \subseteq \mathcal{S}$
    **begin**
        $\mathcal{A} \leftarrow \emptyset$;
        **foreach** $y \in \mathcal{S}$ **do**
1           $\delta_y \leftarrow H(y) - \tilde{H}_\varepsilon(y \mid \mathcal{V} \setminus y)$;
        **for** $j = 1$ **to** $k$ **do**
2           $y^* \leftarrow \arg\max_y \delta_y$;
            $\mathcal{A} \leftarrow \mathcal{A} \cup y^*$;
            **foreach** $y \in N(y^*; \varepsilon)$ **do**
3               $\delta_y \leftarrow \tilde{H}_\varepsilon(y \mid \mathcal{A}) - \tilde{H}_\varepsilon(y \mid \bar{\mathcal{A}})$;
    **end**
```

Algorithm 3: Approximation algorithm for maximizing mutual information using local kernels.

measurement error of at least σ^2. The total decrease of entropy $H(y \mid \tilde{\mathcal{B}}) - H(y \mid \mathcal{B})$ is bounded by $n\varepsilon^2/\sigma^4$. This truncation allows to compute $H(y \mid \bar{\mathcal{A}})$ much more efficiently, at the expense of this small absolute error. In the special case of isotropic kernels, the number d of variables x with $\mathcal{K}(x,y) > \varepsilon$ can be computed as a function of the discretization and the covariance kernel. This reduces the complexity of computing $H(y \mid \bar{\mathcal{A}})$ from $O(n^3)$ to $O(d^3)$, which is a constant.

Our truncation approach leads to the more efficient optimization algorithm shown in Algorithm 3. Here, $\tilde{H}_\varepsilon$ refers to the truncated computation of entropy as described above, and $N(y^*;\varepsilon) \leq d$ refers to the set of elements $x \in \mathcal{S}$ for which $|\mathcal{K}(y^*,x)| > \varepsilon$. Using this approximation, our algorithm is significantly faster: Initialization (Line 1) requires $O(nd^3)$ operations. For each one of the k iterations, finding the next sensor (Line 2) requires $O(n)$ comparisons, and adding the new sensor y^* can only change the score of its neighbors ($N(y^*;\varepsilon) \leq d$), thus Line 3 requires $O(d \cdot d^3)$ operations. The total running time of Algorithm 3 is $O(nd^3 + kn + kd^4)$, which can be significantly lower than the $O(kn^4)$ operations required by Algorithm 1. Theorem 9 summarizes our analysis:

Theorem 9 *Under the assumptions of Lemma 5, guaranteeing ε_1-approximate monotonicity and truncation parameter ε_2, Algorithm 3 selects $\mathcal{A} \subseteq \mathcal{S}$ such that*

$$\mathrm{MI}(\mathcal{A}) \geq (1 - 1/e)(\mathrm{OPT} - k\varepsilon_1 - 2kn\varepsilon_2/\sigma^4),$$

in time $O(nd^3 + nk + kd^4)$.

This approach can be efficiently implemented by using a priority queue to maintain the advantages δ_y. Using for example a Relaxed Heaps data structure, the running time can be decreased to $O(nd^3 + kd\log n + kd^4)$: Line 1 uses the **insert** operation with complexity $O(1)$, Line 2 calls **deletemax** with complexity $O(\log n)$, and Line 3 uses **delete** and **insert**, again with complexity $O(\log n)$. This complexity improves on Algorithm 3 if $d\log n \ll n$. This assumption is frequently met in practice, since d can be considered a constant as the size n of the sensing area grows. Of course, this procedure can also be combined with the lazy evaluations described in the previous section for further improvement in running time.

6. Robust Sensor Placements

In this section, we show how the mutual information criterion can be extended to optimize for placements which are robust against failures of sensor nodes, and against uncertainty in the model parameters. The submodularity of mutual information will allow us to derive approximation guarantees in both cases.

6.1 Robustness Against Failures of Nodes

As with any physical device, sensor nodes are susceptible to failures. For example, the battery of a wireless sensor can run out, stopping it from making further measurements. Networking messages containing sensor values can be lost due to wireless interference. In the following, we discuss how the presented approach can handle such failures. We associate with each location $y_i \in \mathcal{S}$ a discrete random variable Z_i such that $Z_i = 0$ indicates that a sensor placed at location y_i has failed and will not produce any measurements, and $Z_i = 1$ indicates that the sensor is working correctly. For a placement $\mathcal{A} \subset \mathcal{S}$, denote by $\mathcal{A}_{\mathbf{z}}$ the subset of locations $y_i \in \mathcal{A}$ such that $z_i = 1$, that is, the subset of functional sensors. Then, the robust mutual information

$$\mathrm{MI}_R(\mathcal{A}) = \mathbb{E}_{\mathbf{Z}}[\mathcal{A}_{\mathbf{z}}] = \sum_{\mathbf{z}} P(\mathbf{z})\,\mathrm{MI}(\mathcal{A}_{\mathbf{z}}),$$

is an expectation of the mutual information for placement $\mathcal{A}$ where all possible failure scenarios are considered.

Proposition 10 $\mathrm{MI}_R(\mathcal{A})$ *is submodular and, under the assumptions of Lemma 5, approximately monotonic.*

Proof This is a straightforward consequence of the fact that the class of submodular functions are closed under taking expectations. The approximate monotonicity can be verified directly from the approximate monotonicity of mutual information. ■

Unfortunately, the number of possible failure scenarios grows exponentially in $|\mathcal{S}|$. However, if the Z_i are i.i.d., and the failure probability $P(Z_i = 0) = \theta$ is low enough, MI_R can be approximated well, for example, by only taking into account scenarios were none or at most one sensor fails. This simplification often works in practice (Lerner and Parr, 2001). These $|\mathcal{S}| + 1$ scenarios can easily be enumerated. For more complex distributions over Z, or higher failure probabilities θ, one might have to resort to sampling in order to compute MI_R.

The discussion above presents a means for explicitly optimizing placements for robustness. However, we can show that even if we do not specifically optimize for robustness, our sensor placements will be inherently robust:

Proposition 11 (Krause et al., 2006) *Consider a submodular function $F(\cdot)$ on a ground set $\mathcal{S}$, a set $\mathcal{B} \subseteq \mathcal{S}$, and a probability distribution over subsets $\mathcal{A}$ of $\mathcal{B}$ with the property that, for some constant ρ, we have $\Pr[v \in \mathcal{A}] \geq \rho$ for all $v \in \mathcal{B}$. Then $\mathbb{E}[F(\mathcal{A})] \geq \rho F(\mathcal{B})$.*

When applying this proposition, the set $\mathcal{B}$ will correspond to the selected sensor placement. The (randomly chosen) set $\mathcal{A}$ denotes the set of fully functioning nodes. If each node fails independently with probability $1-\rho$, that implies that $\Pr[c \in \mathcal{A}] \geq \rho$, and hence the expected mutual information of the functioning nodes, $\mathbb{E}[\mathrm{MI}(\mathcal{A})]$, is at least ρ times the mutual information $\mathrm{MI}(\mathcal{B})$, that is, when no nodes fail. Proposition 11 even applies if the node failures are not independent, but for example are spatially correlated, as can be expected in practical sensor placement scenarios.

6.2 Robustness Against Uncertainty in the Model Parameters

Often, the parameters θ of the GP prior, such as the amount of variance and spatial correlation in different areas of the space, are not known. Consequently, several researcher (Caselton et al., 1992; Zimmerman, 2006; Zhu and Stein, 2006) have proposed approaches to explicitly address the uncertainty in the model parameters, which are discussed in Section 7.

We want to exploit submodularity in order to get performance guarantees on the placements. We take a Bayesian approach, and equip θ with a prior. In this case, the objective function becomes

$$\mathrm{MI}_M(\mathcal{A}) = \mathbb{E}_\theta[I(\mathcal{A};\mathcal{V}\setminus\mathcal{A} \mid \theta)] = \int p(\theta) I(\mathcal{A};\mathcal{V}\setminus\mathcal{A} \mid \theta) d\theta.$$

Since the class of submodular functions is closed under expectations, MI_M is still a submodular function. However, the approximate monotonicity requires further assumptions. For example, if the discretization meshwidth is fine enough to guarantee approximate monotonicity for all values of θ for which $p(\theta) > 0$, then approximate monotonicity still holds, since

$$\begin{aligned}\mathrm{MI}_M(\mathcal{A}\cup y) - \mathrm{MI}_M(\mathcal{A}) &= \int p(\theta)[I(\mathcal{A}\cup y;\mathcal{V}\setminus(\mathcal{A}\cup y) \mid \theta) - I(\mathcal{A};\mathcal{V}\setminus\mathcal{A} \mid \theta)]d\theta \\ &\geq \int p(\theta)[-\varepsilon]d\theta = -\varepsilon.\end{aligned}$$

A weaker assumption also suffices: If there exists a (nonnegative) function $\eta(\theta)$ such that $I(\mathcal{A}\cup y;\mathcal{V}\setminus(\mathcal{A}\cup y) \mid \theta) - I(\mathcal{A};\mathcal{V}\setminus\mathcal{A} \mid \theta) \geq -\eta(\theta)$, and $\int p(\theta)[-\eta(\theta)]d\theta \geq -\varepsilon$, then MI_M is still ε-approximately monotonic. Such a function would allow the level ε of ε-approximately monotonicity to vary for different values of θ.

Note that in this setting however, the predictive distributions (1) and (2) cannot be computed in closed form anymore, and one has to resort to approximate inference techniques (cf. Rasmussen and Williams, 2006).

The advantage of exploiting submodularity for handling uncertainty in the model parameters is that the offline and online bounds discussed in Section 4.4 still apply. Hence, contrary to existing work, our approach provides strong theoretical guarantees on the achieved solutions.

7. Related Work

There is a large body of work related to sensor placement, and to the selection of observations for the purpose of coverage and prediction. Variations of this problem appear in spatial statistics, active learning, and experimental design. Generally, the methods define an objective function (Section 7.1), such as area coverage or predictive accuracy, and then apply a computational procedure (Section 7.2) to optimize this objective function. We also review related work on extensions to this basic scheme (Section 7.3), the related work in Machine Learning in particular (Section 7.4), and our previous work in this area (Section 7.5).

7.1 Objective Functions

We distinguish *geometric* and *model-based* approaches, which differ according to their assumptions made about the phenomenon to be monitored.

7.1.1 GEOMETRIC APPROACHES

Geometric approaches do not build a probabilistic model of the underlying process, but instead use geometric properties of the space in which the process occurs. The goal is typically a sensor placement that covers the space. The most common approaches for optimizing sensor placements using geometric criteria assume that sensors have a fixed region (cf. Hochbaum and Maas, 1985; Gonzalez-Banos and Latombe, 2001; Bai et al., 2006). These regions are usually convex or even circular. Furthermore, it is assumed that everything within this region can be perfectly observed, and everything outside cannot be measured by the sensors. In Section 8.1, we relate these geometric approaches to our GP-based formulation.

In the case where the sensing area is a disk (the *disk model*), Kershner (1939) has shown that an arrangement of the sensors in the centers of regular hexagons is asymptotically optimal, in the sense that a given set is fully covered by uniform disks. In Section 9.3, we experimentally show that when we apply the disk model to nonstationary placement problems, as considered in this paper, the geometric disk model approach leads to worse placements in terms of prediction accuracy, when compared to model-based approaches.

If many sensors are available then one can optimize the deployment density instead of the placement of individual sensors (Toumpis and Gupta, 2005). The locations of placed sensors are then assumed to be randomly sampled from this distribution. In the applications we consider, sensors are quite expensive, and optimal placement of a small set of them is desired.

7.1.2 MODEL-BASED APPROACHES

This paper is an example of a model-based method, one which takes a model of the world (here, a GP) and places sensors to optimize a function of that model (here, mutual information).

Many different objective functions have been proposed for model-based sensor placement. In the statistics community, classical and Bayesian experimental design focused on the question of selecting observations to maximize the quality of parameter estimates in linear models (cf. Atkinson, 1988; Lindley, 1956). In spatial statistics, information-theoretic measures, notably entropy, have been frequently used (Caselton and Hussain, 1980; Caselton and Zidek, 1984; Caselton et al., 1992; Shewry and Wynn, 1987; Federov and Mueller, 1989; Wu and Zidek, 1992; Guttorp et al., 1992). These objectives minimize the uncertainty in the prediction, after the observations are made.

Classical Experimental Design Criteria. In the statistics literature, the problem of optimal experimental design has been extensively studied (cf. Atkinson, 1988, 1996; Pukelsheim, 1987; Boyd and Vandenberghe, 2004). The problem commonly addressed there is to estimate the parameters θ of a function,

$$y = f_\theta(\mathbf{x}) + w,$$

where w is normally distributed measurement noise with zero mean and variance σ^2, y a scalar output and $\mathbf{x}$ a vectorial input. The assumption is, that the input $\mathbf{x}$ can be selected from a menu of design options, $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Each input corresponds to a possible experiment which can be performed. In our sensor placement case, one $\mathbf{x}$ would be associated with each location, y would be the measurement at the location, and θ would correspond to the values of the phenomenon at the unobserved locations. Usually, the assumption is that f_θ is linear, that is, $y = \theta^T \mathbf{x} + w$.

For the linear model $y = \theta^T \mathbf{x} + w$, if all n observations were available, then

$$\begin{aligned} \mathrm{Var}(\hat{\theta}) &= \sigma^2 (X^T X)^{-1} \\ \mathrm{Var}(\hat{y}_i) &= \sigma^2 \mathbf{x}_i^T (X^T X)^{-1} \mathbf{x}_i, \end{aligned} \tag{10}$$

where X is the design matrix, which consists of the inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$ as its rows. We can see that the variance of both the parameter estimate $\hat{\theta}$ and the predictions $\hat{y}_i$ depends on the matrix $M = (X^T X)^{-1}$, which is called the inverse moment matrix. If this matrix is "small", then the parameter estimates and predictions will be accurate. A design consists of a selection $\mathcal{A}$ of the inputs (with repetitions allowed). We write $X_\mathcal{A}$ to denote the selected experiments, and $M_\mathcal{A}$ for the corresponding inverse moment matrix. Classical experimental design considers different notions of "smallness" for this inverse moment matrix $M_\mathcal{A}$; D-optimality refers to the determinant, A-optimality to the trace and E-optimality to the spectral radius (the maximum eigenvalue). There are several more scalarizations of the inverse moment matrix, and they are commonly referred to as "alphabetical" optimality criteria.

An example of the relationship between this formalism and sensor placements in GPs, as well as experimental comparisons, are presented in Section 9.5.

Equation (10) shows that the distribution of the test data is not taken into account, when attempting to minimizing the inverse moment matrix $M_\mathcal{A}$. Yu et al. (2006) extend classical experimental design to the transductive setting, which takes the distribution of test data into account. The information-theoretic approaches, which we use in this paper, also directly take into account the unobserved locations, as they minimize the uncertainty in the posterior $P(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}})$.

Bayesian Design Criteria. Classical experimental design is a Frequentist approach, which attempts to minimize the estimation error of the maximum likelihood parameter estimate. If one places a prior on the model parameters, one can formalize a Bayesian notion of experimental design. In its general form, Bayesian experimental design was pioneered by Lindley (1956). The users encode their preferences in a utility function $U(P(\Theta), \theta^\star)$, where the first argument, $P(\Theta)$, is a distribution over states of the world (i.e., the parameters) and the second argument, $\theta^\star$, is the true state of the world. Observations $\mathbf{x}_{\mathcal{A}}$ are collected, and the change in expected utility under the prior $P(\Theta)$ and posterior $P(\Theta \mid \mathcal{X}_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}})$ can be used as a design criterion. By using different utility functions, Bayesian versions of *A*-, *D*-, and *E*- optimality can be developed (Chaloner and Verdinelli, 1995). If we have the posterior covariance matrix $\Sigma_{\theta|A}$, whose maximum eigenvalue is $\lambda_{\max}$, then Bayesian *A*-, *D*-, and *E*- optimality minimizes $\operatorname{tr}\left(\Sigma_{\theta|A}\right)$, $\det\left(\Sigma_{\theta|A}\right)$, and $\lambda_{\max}\left(\Sigma_{\theta|A}\right)$, respectively.

Usually, Bayesian experimental design considers the task of parameter estimation (Sebastiani and Wynn, 2000; Paninski, 2005; Ylvisaker, 1975). Lindley (1956) suggested using negative Shannon information, which is equivalent to maximizing the expected Kullback-Leibler divergence between the posterior and prior over the parameters:

$$\int P(\mathbf{x}_{\mathcal{A}}) \int P(\theta \mid \mathbf{x}_{\mathcal{A}}) \log \frac{P(\theta \mid \mathbf{x}_{\mathcal{A}})}{P(\theta)}\, d\theta d\mathbf{x}_{\mathcal{A}}. \tag{11}$$

If we consider distributions $P(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}})$ over the unobserved locations $\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}$ instead of distributions over parameters $P(\Theta)$, (11) leads to the following criterion:

$$\int P(\mathbf{x}_{\mathcal{A}}) \int P(\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathbf{x}_{\mathcal{A}}) \log \frac{P(\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathbf{x}_{\mathcal{A}})}{P(\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}})}\, d\mathbf{x}_{\mathcal{V}\setminus\mathcal{A}} d\mathbf{x}_{\mathcal{A}}. \tag{12}$$

Note that Equation (12) is exactly the mutual information between the observed and unobserved sensors, $I(\mathcal{A}; \mathcal{V}\setminus\mathcal{A})$. For a linear-Gaussian model, where the mean and covariance are known, we get the mutual information criterion of Caselton and Zidek (1984), which we use in this paper.

Information-Theoretic Criteria. The special case of Bayesian experimental design, where an information-theoretic functional (such as entropy or mutual information) is used as a utility function, and where the predictive uncertainty in the unobserved variables is concerned (as in Equation 12) is of special importance for spatial monitoring.

Such information-theoretic criteria have been used as design criteria in a variety of fields and applications. Maximizing the entropy $H(\mathcal{A})$ of a set of observations, as discussed in Section 3, has been used in the design of computer experiments (Sacks et al., 1989; Currin et al., 1991), function interpolation (O'Hagan, 1978) and spatial statistics (Shewry and Wynn, 1987). This criterion is sometimes also referred to as D-optimality, since the scalarization of the posterior variance in the spatial literature and the scalarization of the parameter variance in classical experimental design both involve a determinant. In the context of parameter estimation in linear models and independent, homoscedastic noise, maximizing the entropy $H(\mathcal{A})$ is equivalent to Bayesian D-optimal design (which maximizes the information gain $H(\Theta) - H(\Theta \mid \mathcal{A})$ about the parameters), as discussed by Sebastiani and Wynn (2000) (see also Section 8.4).

Maximizing mutual information between sets of random variables has a long history of use in statistics (Lindley, 1956; Bernardo, 1979), machine learning (Luttrell, 1985; MacKay, 1992). The specific form addressed in this paper, $I(\mathcal{A}; \mathcal{V} \setminus \mathcal{A})$, has been used in spatial statistics (Caselton and Zidek, 1984; Caselton et al., 1992). Mutual information requires an accurate estimate of the joint model $P(\mathcal{X}_{\mathcal{V}})$, while entropy only requires an accurate estimate at the selected locations, $P(\mathcal{X}_{\mathcal{A}})$. Caselton et al. (1992) argue that latter is easier to estimate from a small amount of data, thus arguing against mutual information. We however contend that nowadays effective techniques for learning complex nonstationary spatial models are available, such as the ones used in our experiments, thus mitigating these concerns and enabling the optimization of mutual information.

7.2 Optimization Techniques

All of the criteria discussed thus far yield challenging combinatorial optimization problems. Several approaches are used to solve them in the literature, which can be roughly categorized into those that respect the integrality constraint and those which use a continuous relaxation.

7.2.1 COMBINATORIAL SEARCH

For both geometric and model-based approaches, one must search for the best design or set of sensor locations among a very (usually exponentially) large number of candidate solutions. In a classical design, for example, the inverse moment matrix on a set of selected experiments $\mathcal{X}_{\mathcal{A}}$ can be written as

$$M_{\mathcal{A}} = \left(\sum_{i=1}^{n} k_i \mathbf{x}_i \mathbf{x}_i^T \right)^{-1},$$

where k_i is the number of times experiment $\mathbf{x}_i$ is performed in design $\mathcal{A}$. Since k_i must be an integer, a combinatorial number of potential experimental designs has to be searched. Similarly, when placing a set $\mathcal{A}$ of k sensors out of a set $\mathcal{V}$ of possible locations, as we do in this paper, all sets of size k have to be searched. For both entropy (Ko et al., 1995) and mutual information (this paper), this search has been shown to be NP-hard, hence efficient exact solutions are likely not possible.

Since exhaustive search is usually infeasible, local, heuristic searches without theoretical guarantees have commonly been applied. Approaches to the difficult combinatorial optimization include simulated annealing (Meyer and Nachtsheim, 1988), pairwise exchange (Fedorov, 1972; Mitchell, 1974a,b; Cook and Nachtsheim, 1980; Nguyen and Miller, 1992), forward and backward greedy heuristics (MacKay, 1992; Caselton and Zidek, 1984). All these approaches provide no guarantees about the quality of the solution. Since optimal solutions are highly desirable, branch-and-bound approaches to speed up the exhaustive search have been developed (Welch, 1982; Ko et al., 1995).

Although they enable exhaustive search for slightly larger problem instances, the computational complexity of the problems puts strong limits on their effectiveness.

By exploiting submodularity of mutual information, in this paper, we provide the first approach to information-theoretic sensor placement which has guarantees both on the runtime and on the quality of the achieved solutions.

7.2.2 CONTINUOUS RELAXATION

In some formulations, the integrality constraint is relaxed. For example, in classical experimental design, the number of experiments to be selected is often large compared to the number of design choices. In these cases, one can find a fractional design (i.e., a non-integral solution defining the proportions by which experiments should be performed), and round the fractional solutions. In the fractional formulation, A-, D-, and E-optimality criteria can be solved exactly using a semi-definite program (Boyd and Vandenberghe, 2004). There are however no known bounds on the integrality gap, that is, the loss incurred by this rounding process.

In other approaches (Seo et al., 2000; Snelson and Ghahramani, 2005), a set of locations is chosen not from a discrete, but a continuous space. If the objective function is differentiable with respect to these locations, gradient-based optimization can be used instead of requiring combinatorial search techniques. Nevertheless, optimality of the solution is not guaranteed since there is no known bound on the discrepancy between local and global optima.

Another method that yields a continuous optimization, in the case of geometric objective functions, is the potential field approach (Heo and Varshney, 2005; Howard et al., 2002). An energy criterion similar to a spring model is used. This optimization results in uniformly distributed (in terms of inter-sensor distances), homogeneous placements. The advantage of these approaches is that they can adapt to irregular spaces (such as hallways or corridors), where a simple grid-based deployment is not possible. Since the approach uses coordinate ascent, it can be performed using a distributed computation, making it useful for robotics applications where sensors can move.

7.3 Related Work on Extensions

In this section, we discuss prior work related to our extensions on sensor placement under model uncertainty (Section 6) and on the use of non-constant cost functions (Section 4.3).

7.3.1 PLACEMENT WITH MODEL UNCERTAINTY

The discussion thus far has focused on the case where the joint model $P(\mathcal{X}_{\mathcal{V}})$ is completely specified, that is, the mean and covariance of the GP are known.[2] With model uncertainty, one has to distinguish between observation selection for predictive accuracy in a fixed model and observation selection for learning parameters. Model uncertainty also introduces computational issues. If the mean and covariance are fixed in a Gaussian process then the posterior is Gaussian. This makes it

[2] Or one assumes the uncertainty on these parameters is small enough that their contribution to the predictive uncertainty is negligible.

easy to compute quantities such as entropy and mutual information. If the mean and covariance are unknown, and we have to learn hyperparameters (e.g., kernel bandwidth of an isotropic process), then the predictive distributions and information-theoretic quantities often lack a closed form.

Caselton et al. (1992) extend their earlier work on maximum entropy sampling to the case where the mean and covariance are unknown by using a conjugate Bayesian analysis. The limitations of this approach are that the conjugate Bayesian analysis makes spatial independence assumptions in the prior and that complete data with repeated observations are required at every potential sensing site. This leads to a determinant maximization problem, much like D-optimality, that precludes the use of submodularity.

Another approach is the development of hybrid criteria, which balance parameter estimation and prediction. For example, Zimmerman (2006) proposes local EK-optimality, a linear combination of the maximum predictive variance and a scalarization of the covariance of the maximum likelihood parameter estimate. While this criterion selects observations which reduce parameter uncertainty and predictive uncertainty given the *current parameter*, it does not take into account the effect of parameter uncertainty on prediction error. To address this issue, Zhu and Stein (2006) derive an iterative algorithm which alternates between optimizing the design for covariance estimation and spatial prediction. This procedure does not provide guarantees on the quality of designs.

An alternative approach to addressing model uncertainty, in the context of classical experimental design, is presented by Flaherty et al. (2006). There, instead of committing to a single value, the parameters of a linear model are constrained to lie in a bounded interval. Their robust design objective, which is based on E-optimality, is then defined with respect to the worst-case parameter value. Flaherty et al. (2006) demonstrate how a continuous relaxation of this problem can be formulated as a SDP, which can be solved exactly. No guarantees are given however on the integrality gap on this relaxation.

In our approach, as discussed in Section 6, we show how submodularity can be exploited even in the presence of parameter uncertainty. We do not address the computational issues, which depend on the particular parameterization of the GP used. However, in special cases (e.g., uncertainty about the kernel bandwidth), one can apply sampling or numerical integration, and still get guarantees about the achieved solution.

7.3.2 NON-CONSTANT COST FUNCTIONS

In Section 4.3, we discuss the case where every sensor can have a different cost, and one has a budget which one can spend. An alternate approach to sensor costs is presented by Zidek et al. (2000). They propose a criterion that makes a trade off between achieved reduction in entropy using an entropy-to-cost conversion factor, that is, they optimize the sum of the entropy with a factor times the cost of the placements. This criterion yields an unconstrained optimization problem. Our approach to sensor costs (Section 4.3) yields a constrained optimization, maximizing our criteria given a fixed budget that can be spent when placing sensors. Such a budget-based approach seems more natural in real problems (where one often has a fixed number of sensors or amount of money to spend). Moreover, our approach provides strong a priori theoretical guarantees and tighter online bounds, which are not available for the approach of Zidek et al. (2000).

7.4 Related Work in Machine Learning

In Machine Learning, several related techniques have been developed for selecting informative features, for active learning and for speeding up GP inference.

7.4.1 FEATURE SELECTION AND DIMENSION REDUCTION

Given that the joint distribution of $\mathcal{X}_\mathcal{A}$ and $\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}$ is Gaussian, their mutual information is also

$$\mathrm{MI}(\mathcal{A}) = -\frac{1}{2}\sum_i \log\left(1-\rho_i^2\right) \tag{13}$$

where $\rho_1^2 \geq \cdots \geq \rho_{|\mathcal{V}|}^2$ are the canonical correlation coefficients between $\mathcal{X}_\mathcal{A}$ and $\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}$ (Caselton and Zidek, 1984). McCabe (1984) show that maximizing the canonical correlations between observed and unobserved variables can be interpreted as a form of principal components analysis, where one realizes that selecting subsets of variables is a special kind of linear projection. A similar analysis is presented for entropy and other common design criteria. Using Equation (13), a similar relationship can be made to canonical correlation analysis (CCA; Hotelling, 1936), which finds linear projections for $\mathcal{V}\setminus\mathcal{A}$ and $\mathcal{A}$ that maximize the correlations in the lower dimensional space. By considering these lower-dimensional projections, one can determine how much variance is shared (jointly explained) by $\mathcal{V}\setminus\mathcal{A}$ and $\mathcal{A}$.

While dimension reduction techniques such as Principal Component Analysis (PCA) or CCA can be used to find a lower dimensional representation of a high dimensional problem, these techniques usually find projections which are non-sparse, that is, which are linear combinations of (almost) all input variables. However, for interpretation purposes (and considering data acquisition cost), one often desires *sparse* projections, which are linear combinations of only a small subset of input variables. Moghaddam et al. (2005) and Moghaddam et al. (2006) consider the problem of selecting such sparse linear projections (subject to a constraint on the number of nonzero entries) of minimum reconstruction error (for PCA) and class separation (for LDA). In order to find these sparse projections, they propose two approaches: A mixed integer program, which can solve the problem optimally—albeit generally not in polynomial time, and a heuristic approach, using a greedy forward search followed by a greedy backward elimination. They provide several theoretical bounds, including a guarantee that this backward greedy algorithm achieves a solution of at least $\frac{k}{n}\lambda_{\max}$ where $n = |\mathcal{V}|$, and k is the number of chosen observations (Moghaddam, 2007).

7.4.2 ACTIVE LEARNING

In the machine learning community, information-theoretic criteria have been used for active learning, techniques which allow the learning algorithm to influence the choice of training samples. For example, information-theoretic criteria have been used in the analysis of query-by-committee to select samples (Sollich, 1996; Freund et al., 1997; Axelrod et al., 2001). Following Lindley (1956), MacKay (1992) proposes selecting observations that maximize expected information gain, either in terms of entropy or cross entropy, using Federov exchange. As opposed to this paper, which addresses the optimization problem, MacKay (1992) focuses on comparing the different objective criteria. Cohn (1994) proposes scoring each potential observation by measuring the average reduction in predicted variance at a set of reference points. There is some evidence which suggests that this approach can improve prediction in Gaussian process regression (Seo et al., 2000). Common to all these active learning approaches, as well as to this paper, is the problem of selecting a set of most informative observation. Unlike this paper, we are not aware of any prior work in this area which provides rigorous approximation guarantees for this problem.

7.4.3 FAST GAUSSIAN PROCESS METHODS

Information-theoretic criteria are also used in sparse GP modeling, which attempts to reduce the cost of inference by selecting a representative subset of the training data. Sample selection criteria have included KL-divergence (Seeger et al., 2003) and entropy (Lawrence et al., 2003). In contrast to sensor placement, where locations are chosen to minimize predictive uncertainty, in sparse GP methods, the samples are chosen such that the approximate posterior matches the true posterior (which uses the entire training set) as accurately as possible. Instead of choosing a subset of the training data, Snelson and Ghahramani (2005) propose to optimize the location of a set of “hallucinated” inputs. This approach results in a continuous optimization problem, which appears to be easier to solve (albeit with no performance guarantees) than the discrete subset selection problem.

7.5 Relationship to Previous Work of the Authors

An earlier version of this paper appeared as (Guestrin et al., 2005). The present version is substantially extended by new experiments on nonstationarity (Section 9.3, Section 9.2) and comparisons to classical experimental design (Section 9.5). New are also the discussion of robust placements in Section 6 and several extensions in Section 4 and Section 5.

Additionally, Krause et al. (2006) presented an approximation algorithm for optimizing node placements for sensor networks using GPs that takes into account both the informativeness of placements (analogously to the discussion in this paper) and the communication cost required to retrieve these measurements. Their approach uses GPs both for modeling the monitored phenomenon as well as the link qualities of the sensor network. Singh et al. (2007) consider the case of planning informative paths for multiple robots. Here, the goal is to select observations which are both informative, but also lie on a collection of paths, one for each robot, of bounded length. They develop an approximation algorithm with theoretical guarantees on the quality of the solution. In the setting of Krause et al. (2006) and Singh et al. (2007)—unlike the case considered in this paper, where there are no constraints on the location of the sensors—the greedy algorithm performs arbitrarily badly, and the papers describe more elaborate optimization algorithms. In these algorithms, the submodularity of mutual information is again the crucial property which allows the authors to obtain approximation guarantees for their approach.

8. Notes on Optimizing Other Objective Functions

In this section, we discuss some properties of alternative optimality criteria for sensor placement.

8.1 A Note on the Relationship with the Disk Model

The disk model for sensor placement (cf. Hochbaum and Maas, 1985; Bai et al., 2006) assumes that each sensor can perfectly observe everything within a radius of r and nothing else. Hence we can associate with every location $y \in \mathcal{V}$ a sensing region D_y, which, for a discretization of the space, corresponds to the locations contained within radius r of location y. For a set of locations $\mathcal{A}$, we can define the coverage $F_D(\mathcal{A}) = \bigcup_{y \in \mathcal{A}} D_y$. It can be easily seen that this criterion is monotonic, submodular and $F(\emptyset) = 0$. Hence optimizing placements for the disk model criterion is a submodular maximization problem, and the greedy algorithm can guarantee a constant factor $(1-1/e)\,\mathrm{OPT}$ approximation guarantee for finding the placement of k sensors with maximum coverage.

There is a sense, in which the approach of sensor placements in GPs can be considered a generalization of the disk model. If we assume an isotropic GP with local kernel function as the one presented in Figure 8, then a sensor measurement is correlated exactly with the locations within a disk around its location. If the process has constant variance, then the greedy algorithm will, for the first few sensors placed, only try to achieve a disjoint placement of the disks, and as such behave just like the greedy algorithm for disk covering.

However, once enough sensors have been placed so that these “disks” start to overlap, the behavior of the two approaches begins to differ: in a disk model there is no advantage in placing sensors that lead to overlapping disks. In a GP model, even an isotropic one, “overlapping disks” lead to better predictions in the overlapping area, a very natural consequence of the representation of the uncertainty in the process.

8.2 A Note on Maximizing the Entropy

As noted by Ko et al. (1995), entropy is also a submodular set function, suggesting a possible application of the result of Nemhauser et al. (1978) to the entropy criterion. The corresponding greedy algorithm adds the sensor y maximizing $H(\mathcal{A} \cup y) - H(\mathcal{A}) = H(y \mid \mathcal{A})$. Unfortunately, our analysis of approximate monotonicity does not extend to the entropy case: Consider $H(y \mid \mathcal{A})$ for $\mathcal{A} = \{z\}$, for sufficiently small measurement noise σ^2, we show that $H(y \mid \mathcal{A})$ can become arbitrarily negative as the mesh width of the discretization decreases. Thus, (even approximate) monotonicity does not hold for entropy, suggesting that the direct application of the result of Nemhauser et al. (1978) is not possible. More precisely, our negative result about the entropy criterion is:

Remark 12 *Under the same assumptions as in Lemma 5, for any $\varepsilon > 0$, there exists a mesh discretization width $\delta > 0$ such that for any discretization level δ', where $0 < \delta' \leq \delta$, entropy violates the monotonicity criterion by at least ε, if $\sigma^2 < \frac{1}{4\pi e}$.*

8.3 A Note on Maximizing the Information Gain

Another objective function of interest is the *information gain* of a sensor placement with respect to some distinguished variables of interest $\mathcal{U}$, that is, $\mathrm{IG}(\mathcal{A}) = I(\mathcal{A}; \mathcal{U}) = H(\mathcal{U}) - H(\mathcal{U} \mid \mathcal{A})$. Unfortunately, this objective function is *not* submodular, even in the case of multivariate normal distributions: Let $\mathcal{X} = (\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)$ be distributed according to a multivariate Gaussian with zero mean and covariance

$$\Sigma = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}.$$

Let $\mathcal{U} = \{X_3\}$. Here, X_2 and X_3 are marginally independent. Thus, alone, X_2 provides no gain in information. However, X_2 does provide information about X_1. Thus, if we first place a sensor at position 1, placing a sensor at position 2 does help. More formally, $\mathrm{IG}(\{\mathcal{X}_2\}) - \mathrm{IG}(\emptyset) = H(\mathcal{X}_3) - H(\mathcal{X}_3 \mid \mathcal{X}_2) < H(\mathcal{X}_3 \mid \mathcal{X}_1) - H(\mathcal{X}_3 \mid \mathcal{X}_1, \mathcal{X}_2) = \mathrm{IG}(\{\mathcal{X}_1, \mathcal{X}_2\}) - \mathrm{IG}(\{\mathcal{X}_1\})$. Hence adding $\mathcal{X}_2$ to the empty set achieves strictly less increase in information gain than adding $\mathcal{X}_2$ to the singleton set containing $\mathcal{X}_1$, contradicting the submodularity assumption.

Remark 13 *The information gain,* $\mathrm{IG}(\mathcal{A}) = I(\mathcal{A}; \mathcal{U})$ *is not submodular in* $\mathcal{A}$.

8.4 A Note on Using Experimental Design for Sensor Placement

As discussed in Section 7.1, the goal of classical experimental design is to find a set $\mathcal{A}$ of experimental stimuli $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ such that the parameter estimation error covariance $M_{\mathcal{A}} = (X_{\mathcal{A}}^T X_{\mathcal{A}})^{-1}$ is as small as possible, where $X_{\mathcal{A}}$ is a matrix containing the $\mathbf{x}_i$ as rows. The different optimality criteria vary in the criteria used for measuring the smallness of $M_{\mathcal{A}}$. Consider the case where the number p of parameters θ is greater than 1, as in the sensor placement setting, where θ are the uninstrumented locations. If we select less than p observations, that is, $|\mathcal{A}| \leq p$, then $(X_{\mathcal{A}}^T X_{\mathcal{A}})$ is not full rank, and $M_{\mathcal{A}}$ is infinite. Hence, for $|\mathcal{A}| < p$, all alphabetical optimality criteria are infinite. Consequently, the A-, D- and E-optimality criteria are infinite as well for all discrete designs $\mathcal{A}$ of size less than p, and hence two such designs are incomparable under these criteria. This incomparability implies that the greedy algorithm will have no notion of improvement, and cannot be used for optimizing discrete classical experimental designs. Hence, discrete classical design cannot be addressed using the concept of submodularity.[3]

In the case of Bayesian experimental design, the parameters are equipped with a prior, and hence the posterior error covariance will not be infinite. In this case however, none of Bayesian A-, D- and E-optimality can be optimized using the result by Nemhauser et al. (1978) in general:

- **Bayesian A-optimality.** Let $y = \theta^T X + w$ where $X = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. Also let $\theta = (\theta_1, \theta_2)$ as well as the noise $w = (w_1, w_2, w_3)$ have independent normal priors with mean 0 and variance 1. Then the joint distribution of (y, θ) is multivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} 3 & 3 & 3 & 1 & 1 \\ 3 & 6 & 4 & 1 & 2 \\ 3 & 4 & 6 & 2 & 1 \\ 1 & 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 0 & 1 \end{bmatrix}.$$

[3] Note that generally, GPs are infinite-dimensional objects, so using classical experimental design for finite linear models only makes limited sense for sensor placement, for example, if an appropriate discretization is chosen.

The goal of A-optimality is to maximally reduce the variance in the posterior distribution over θ, that is, we want to select $\mathcal{A} \subseteq \mathcal{V} = \{y_1, y_2, y_3\}$ in order to maximize $F(\mathcal{A}) = \mathrm{tr}(\Sigma_\theta) - \mathrm{tr}(\Sigma_{\theta|\mathcal{A}})$. Now, we can verify that $F(\{y_1, y_2, y_3\}) - F(\{y_1, y_2\}) \geq F(\{y_1, y_3\}) - F(\{y_1\})$, contradicting submodularity.

- **Bayesian D-optimality.** As shown, for example, by Sebastiani and Wynn (2000), in the case of Bayesian experimental design for linear regression with independent, homoscedastic noise, D-optimality is actually equivalent to the entropy criterion, that is, $\mathrm{argmax}_{|\mathcal{A}| \leq k} I(\Theta, \mathcal{A}) = \mathrm{argmax}_{|\mathcal{A}| \leq k} H(\mathcal{A})$. Hence, the same counterexample as in Section 8.2 shows, that, while Bayesian D-optimality is submodular in this case, it is arbitrarily non-monotonic, and hence the result of Nemhauser et al. (1978) does not apply. In the more general case of dependent, heteroscedastic noise, D-optimality is equivalent to the information gain criterion with $\mathcal{U} = \{\theta\}$ and the counterexample of Section 8.3 applies, contradicting submodularity.

- **Bayesian E-optimality.** Consider the case where $\theta = (\theta_1, \theta_2)$ equipped with a normal prior with zero mean and covariance $diag([1,1])$. Let $y_i = \theta^T \mathbf{x}_i + w$ where w is Gaussian noise with mean zero and variance ε. Let $\mathbf{x}_1 = (1,0)^T$ and $\mathbf{x}_2 = (0,1)^T$. In this setting, the goal of E-optimality is to maximize $F(\mathcal{A}) = \lambda_{\max}(\Sigma_\theta) - \lambda_{\max}(\Sigma_{\theta|\mathcal{A}})$. If we perform no experiment ($\mathcal{A} = \emptyset$), then the posterior error covariance, $\Sigma_{\theta|\mathcal{A}} = \Sigma_\theta$, and hence the maximum eigenvalue is 1. If we observe either y_1 or y_2, the largest eigenvalue is still $\lambda_{\max}(\Sigma_{\theta|\mathcal{X}_\mathcal{A}}) = 1$, and hence $F(\emptyset) = F(\{y_1\}) = F(\{y_2\}) = 0$. But if we observe both y_1 and y_2, then $\lambda_{\max}(\Sigma_{\theta|\mathcal{A}}) = \frac{\varepsilon}{1+\varepsilon}$, and $F(\{y_1, y_2\}) > 0$. Hence $F(\{y_2, y_1\}) - F(\{y_1\}) > F(\{y_2\}) - F(\emptyset)$, that is, adding y_2 helps more if we add it to y_1 than if we add it to the empty set, contradicting submodularity.

Remark 14 *The analysis of Nemhauser et al. (1978) applies to neither of Bayesian A-, D-, and E-optimality in general.*

However, Das and Kempe (2008) show that in certain cases, under a condition of *conditional suppressor-freeness*, the variance reduction F (and hence Bayesian A-optimality) can indeed be shown to be submodular.

9. Experiments

We performed experiments on two real-world data sets, which are described in Section 9.1. In Section 9.2 we compare placements on stationary and nonstationary GP models. In Sections 9.3, 9.4 and 9.5 we compare mutual information with the disk model, with entropy and with other classical experimental design criteria, in terms of prediction accuracy. In 9.6 we compare the performance of the greedy algorithm with other heuristics, and in Section 9.7 we analyze the effect of exploiting local kernels.

9.1 Data Sets

We first introduce the data sets we consider in our experiments. In our first data set, we analyze temperature measurements from the network of 46 sensors shown in Figure 1(a). Our training data consisted of samples collected at 30 sec. intervals on 3 consecutive days (starting Feb. 28th 2004), the testing data consisted of the corresponding samples on the two following days.

Our second data set consists of precipitation data collected during the years 1949 - 1994 in the states of Washington and Oregon (Widmann and Bretherton, 1999). Overall 167 regions of equal area, approximately 50 km apart, reported the daily precipitation. To ensure the data could be reasonably modeled using a Gaussian process we applied a log-transformation, removed the daily mean, and only considered days during which rain was reported. After this preprocessing, we selected the initial two thirds of the data as training instances, and the remaining samples for testing purposes. From the training data, we estimated the mean and empirical covariance, and regularized it by adding independent measurement noise[4] of $\sigma^2 = 0.1$.

We computed error bars for the prediction accuracies in all of our experiments, but due to the violated independence of the collected samples (which are temporally correlated), these error bars are overconfident and hence not reported here. The estimated standard errors under the independence assumption are too small to be visible on the plots.

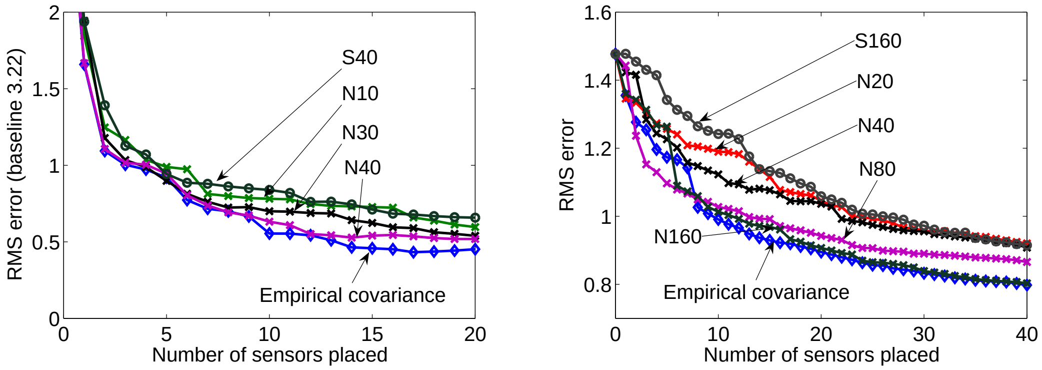
9.2 Comparison of Stationary and Non-stationary Models

To see how well both the stationary and nonstationary models capture the phenomenon, we performed the following experiment: We learned both stationary and non-stationary GP models from an increasing number of sensors. The model with stationary correlation function used an isotropic Exponential kernel with bandwidth fitted using least-squares fit of the empirical variogram (Cressie, 1991). We also learned a nonstationary GP using the technique from Nott and Dunsmuir (2002). Both GP models were estimated from an initial deployment of an increasing number of sensors. We used non-stationary the same variance process for both stationary and nonstationary models (i.e., giving more information to the stationary model than commonly done). Since with increasing amount of data, the empirical covariance matrix will exactly capture the underlying process, we consider the empirical covariance as the ground truth both for placements and prediction. Hence we also selected placements using the entire estimated covariance matrix.

We optimized the designs using mutual information on all the models. We evaluate the prediction accuracy for an increasing number of near-optimally placed sensors, using the estimated model and the measured values for the selected sensors. Figure 9(a) presents the results for the temperature data set. We can see that the nonstationary model learned from 10 sensors performs comparably to the stationary model with 40 sensors, even with non-stationary variance process. As we increase the number of sensors in the initial deployment, the Root Mean Squared error (RMS) prediction accuracies we get for placements of increasing size converge to those obtained for optimizing the placements based on the empirical covariance matrix.

Figure 9(b) presents the results of the same experiment for the precipitation data. Here we can see that the nonstationary model estimated using 20 sensors leads to better RMS accuracies than the stationary model, even if latter is estimated using 160 sensors.

[4] The measurement noise σ^2 was chosen by cross-validation.



(a) *Comparison of fits for temperature data.*

(b) *Comparison of fits for precipitation data.*

Figure 9: RMS curves for placements of increasing size, optimized using stationary and nonstationary GPs. Prediction is done using estimated models. (a) Stationary GP estimated for the temperature data from 40 sensors (S40), nonstationary GPs estimated from 10, 30 and 40 sensors (N10, N30, N40). (b) Stationary GP estimated for the precipitation data from 160 sensors (S160), nonstationary GPs estimated from 20, 40, 80 and 160 sensors (N20, N40, N80, N160).

9.3 Comparison of Data-driven Placements with Geometric Design Criteria

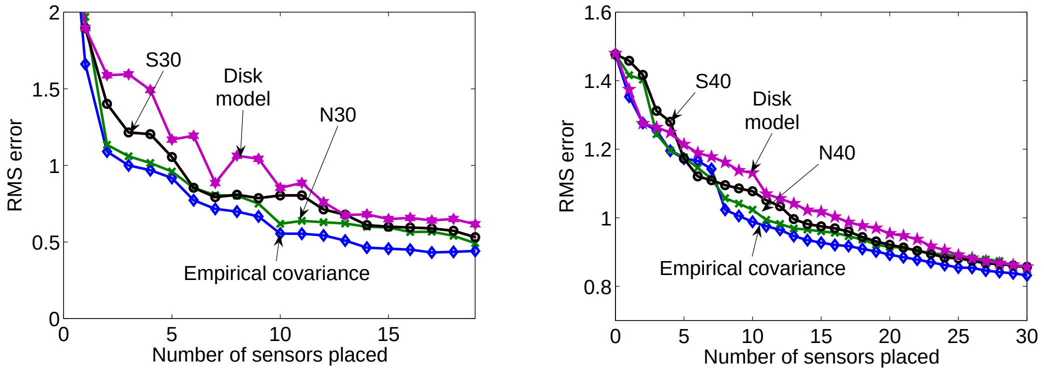
We now compare placements based on our data-driven GP models with those based on the traditional disk model. This model assumes that every sensor can perfectly measure the phenomenon within a radius of r, and have no information outside this radius. Since choosing an appropriate radius for the disk model is very difficult in practice, we decided to choose $r = 5m$ since for this radius 20 sensors could just spatially cover the entire space. We also learned stationary and non-stationary GP models as discussed in Section 9.2.

For the disk model, we used the greedy set covering algorithm. The design on both GP models was done using our greedy algorithm to maximize mutual information. For an increasing number of sensors, we compute the Root Mean Squares (RMS) prediction error on the test data. In order to separate the placement from the prediction task, we used the empirical covariance matrix estimated from the training data on all 46 locations for prediction, for all three placements.

Figure 10(a) presents the results of this experiment. We can see that the geometrical criterion performs poorly compared to the model based approaches. We can see that the placements based on the empirical covariance matrix perform best, quite closely followed by the accuracies obtained by the designs based on the nonstationary process. Figure 10(b) shows the results for the same experiment on the precipitation data set.

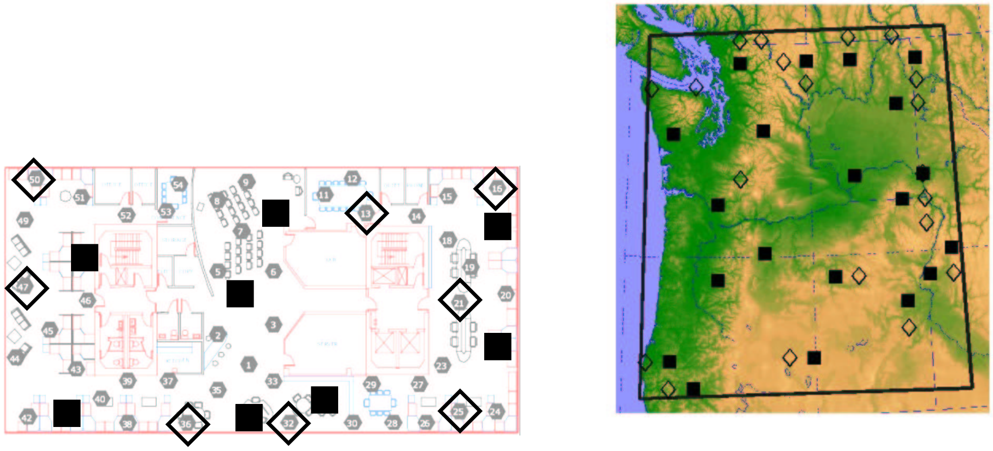
9.4 Comparison of the Mutual Information and Entropy Criteria

We also compared the mutual information criterion to other design criteria. We first compare it against the entropy (variance) criterion. Using the empirical covariance matrix as our process, we use the greedy algorithm to select placements of increasing size, both for mutual information and for entropy. Figure 12(a) and Figure 12(b) show the results of this comparison on models estimated



(a) *Comparison with disk model (Temperature)* (b) *Comparison with disk model (Precipitation)*

Figure 10: RMS curves for placements of increasing size, optimized using the disk model, stationary and nonstationary GPs. Prediction for all placements is done using the empirical covariance. Stationary GPs and nonstationary GPs estimated from 30 sensors (N30, S30, for temperature data) or 40 sensors (N40, S40, for precipitation data).

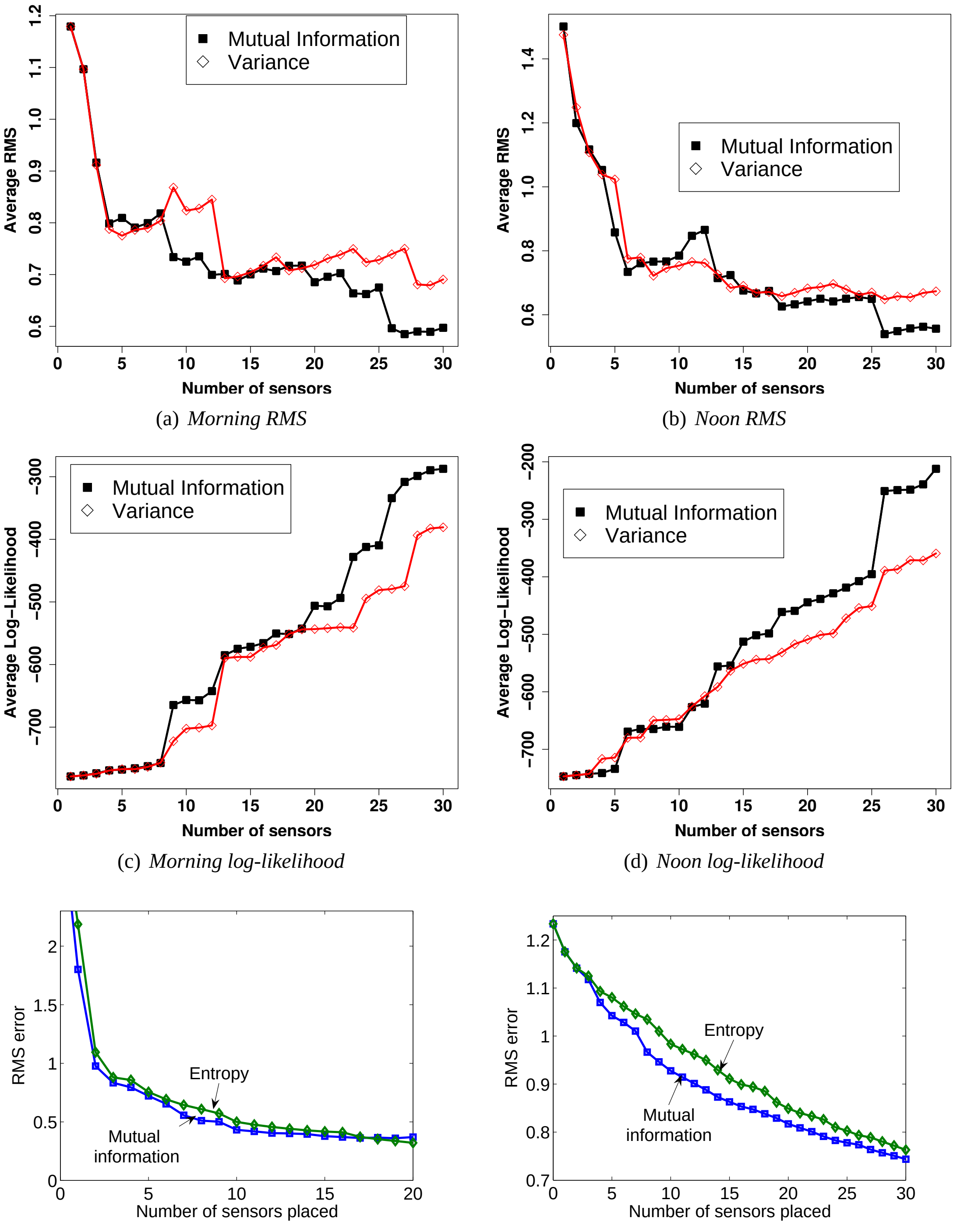


(a) *Placements of temperature sensors* (b) *Placements of rain sensors*

Figure 11: Example sensor placements for temperature and precipitation data. Squares indicate locations selected by mutual information, diamonds indicate those selected by entropy. Notice how entropy places sensors closer to the border of the sensing field.

for the morning (between 8 am and 9 am) and noon (between 12 pm and 1 pm) in the Intel lab data. Figure 12(a) and Figure 12(b) plot the log-likelihood of the test set observations with increasing number of sensors for both models. Figure 12(e) presents the RMS error for a model estimated for the entire day. We can see that mutual information in almost all cases outperforms entropy, achieving better prediction accuracies with a smaller number of sensors.

Figure 12(f) presents the same results for the precipitation data set. Mutual information significantly outperforms entropy as a selection criterion—often several sensors would have to be additionally placed for entropy to reach the same level of prediction accuracy as mutual information. Figure 11(b) shows where both objective values would place sensors to measure precipitation. It can be seen that entropy is again much more likely to place sensors around the border of the sensing area than mutual information.



(a) *Morning RMS*

(b) *Noon RMS*

(c) *Morning log-likelihood*

(d) *Noon log-likelihood*

(e) *Entire day*

(f) *Precipitation RMS*

Figure 12: Prediction error and log-likelihood on test data for temperatures (a-e) and precipitation (f) in sensor network deployment, for an increasing number of sensors.

To gain further insight into the qualitative behavior of the selection criteria we learned a GP model using all sensors over one hour starting at noon. The model was fit with a isotropic Gaussian kernel and quadratic trend for the mean, using the *geoR* Toolkit (Ribeiro Jr. and Diggle, 2001). Figures 13(a) and 13(b) show the posterior mean and variance for the model. Using our algorithms, 22 sensors were chosen using the entropy and mutual information criteria. For each set of selected sensors, additional models were trained using only the measurements of the selected sensors. Predicted temperature surfaces for the entropy and mutual information configurations are presented in Figures 13(c) and 13(d). Entropy tends to favor placing sensors near the boundary as observed in Section 3, while mutual information tends to place the sensors on the top and bottom sides, which exhibited the most complexity and should have a higher sensor density. The predicted variances for each model are shown in figures 13(e) and 13(f). The mutual information version has significantly lower variance than the entropy version almost everywhere, displaying, as expected, higher variance in the unsensed areas in the center of the lab.

9.5 Comparison of Mutual Information with Classical Experimental Design Criteria

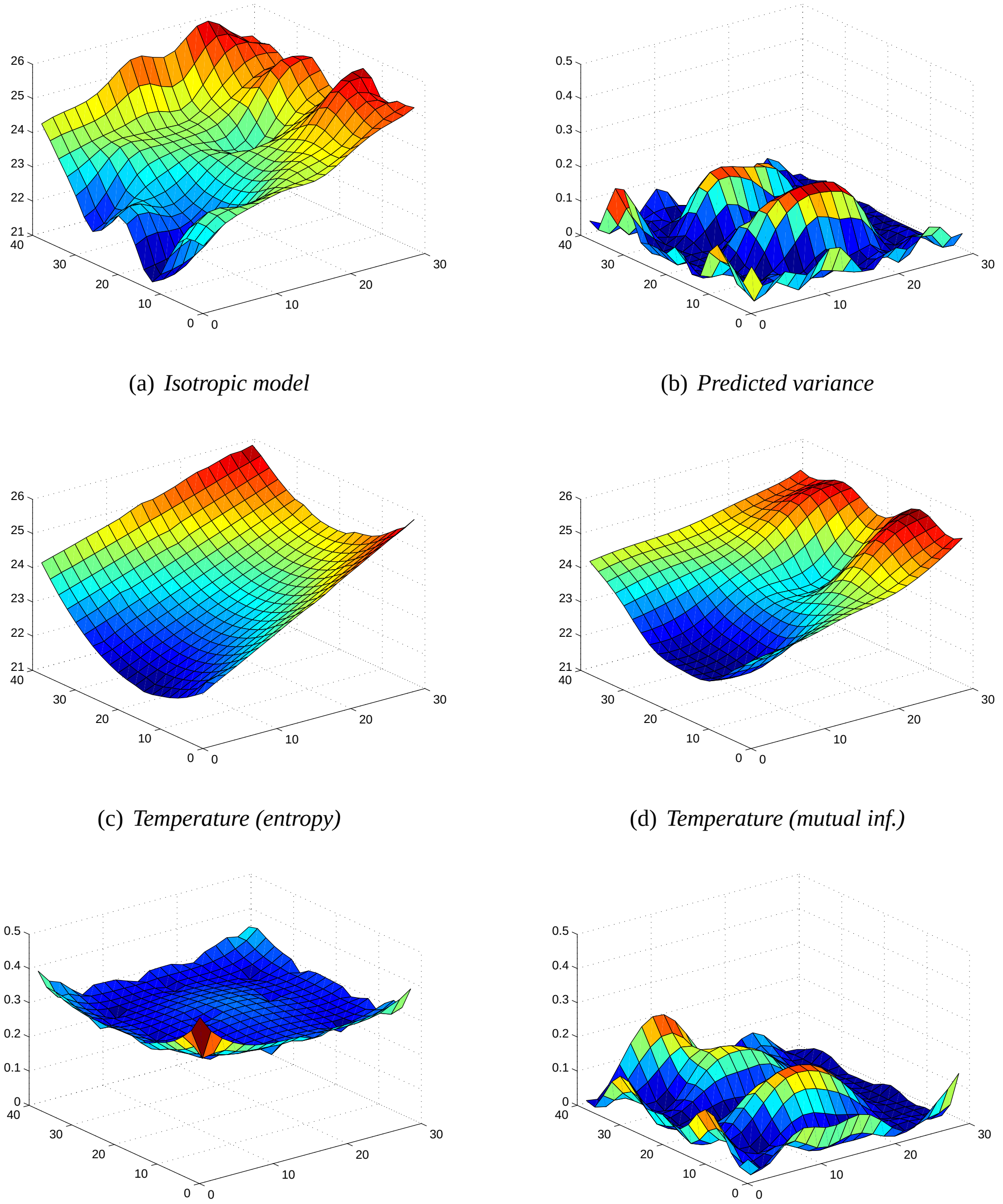
In order to compare the mutual information placements with the classical optimality criteria, we performed the following experiment. We uniformly selected 12 target locations $\mathcal{U}$ in the lab as locations of interest. We then set up the linear model

$$\mathbf{x}_{\mathcal{S}} = \Sigma_{\mathcal{S}\mathcal{U}}\Sigma_{\mathcal{U}\mathcal{U}}^{-1}\,\mathbf{x}_{\mathcal{U}} + \mathbf{w}.$$

Hereby, $\mathbf{x}_{\mathcal{S}}$ denotes measurements at the locations $\mathcal{S}$, among which we choose our placement, $\mathbf{x}_{\mathcal{U}}$ are the values at the locations of interest (no sensors can be placed there), and $\mathbf{w}$ models independent normal measurement noise with constant variance. After subtraction of the training set mean, this model uses the Best Linear Unbiased (Kriging) estimator for predicting $\mathbf{x}_{\mathcal{S}}$ from $\mathbf{x}_{\mathcal{U}}$.

The problem becomes to select the sensor locations $\mathcal{A} \subset \mathcal{S}$ which allow most precise prediction of the variables of interest, in the sense of minimizing the error covariance $\frac{1}{\sigma^2}(A^T A)^{-1}$, where $A = \Sigma_{\mathcal{S}\mathcal{U}}\Sigma_{\mathcal{U}\mathcal{U}}^{-1}$. The different classical design criteria vary in how the scalarization of the error covariance is done. D-optimal design minimizes the log-determinant, A-optimal design minimizes the trace, and E-optimal design minimizes the spectral radius (the magnitude of the largest eigenvalue) of the error covariance. Note that this problem formulation favors the classical design criteria, which are tailored to minimize the error of predicting the values at the target locations $\mathcal{U}$, whereas mutual information and entropy just try to decrease the uncertainty in the entire space.

In order to solve the classical experimental design problems, we use the formulations as a semidefinite program (SDP) as discussed by Boyd and Vandenberghe (2004). We use SeDuMi (Sturm, 1999) for solving these SDPs. Since the integral optimization is hard, we solve the SDP relaxation to achieve a fractional design. This fractional solution defines the best way to distribute an infinite (or very large) budget of experiments to the different choices on the design menu (the variables in $\mathcal{S}$). In the sensor selection problem however, we have to solve the integral problem, since we face the binary decision of whether a sensor should be placed at a particular location or not. This is a hard combinatorial optimization problem. Since no near-optimal solution is known, we select the locations corresponding to the top k coefficients of the design menu, as is common practice. We compare the placements using the classical design criteria to those using the mutual information and entropy criteria, and evaluate each of them on the RMS prediction accuracy on the hold-out locations $\mathcal{U}$.



(a) *Isotropic model*

(b) *Predicted variance*

(c) *Temperature (entropy)*

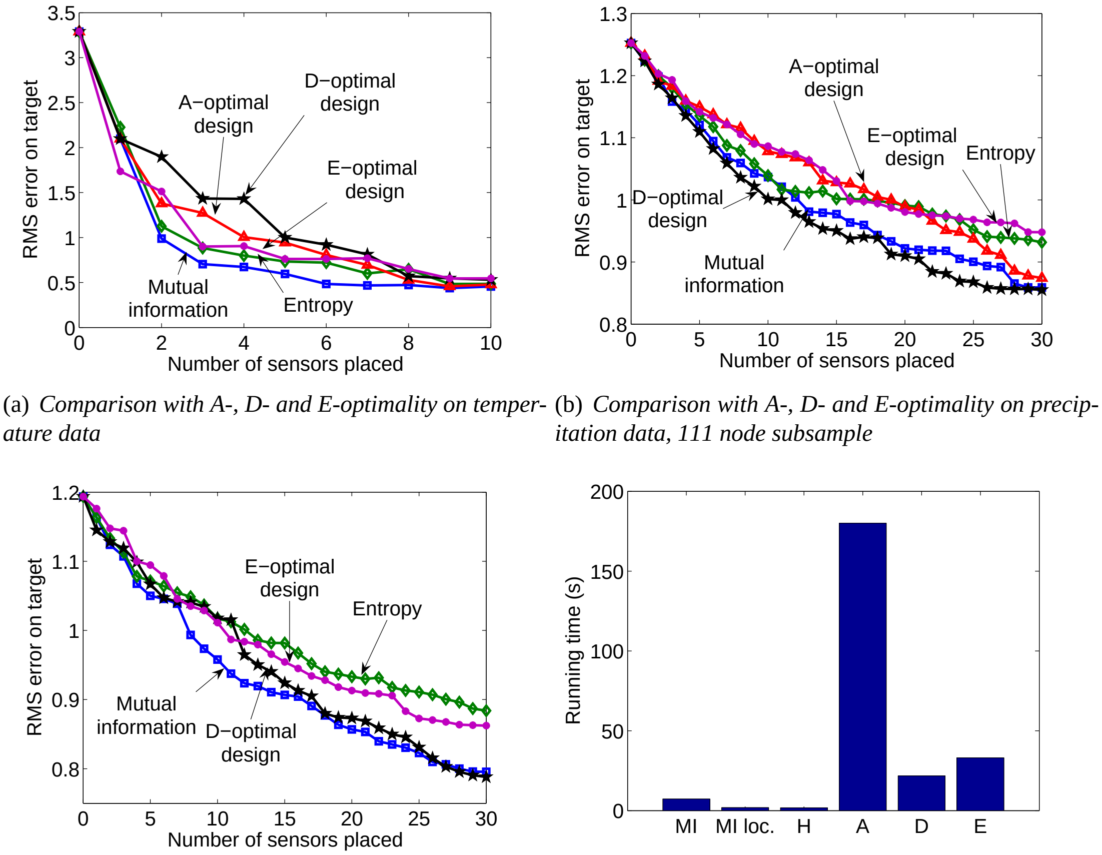
(d) *Temperature (mutual inf.)*

(e) *Variance (entropy)*

(f) *Variance (MI)*

Figure 13: Comparison of predictive quality of subsets selected using MI and entropy.

Figure 14(a) presents the results of this experiment on the temperature data. We can see that even though mutual information optimizes for prediction accuracy in the entire space and not specifically for the target locations $\mathcal{U}$, it incurs the least RMS prediction error, apart from the placements consisting only of a single sensor. E-optimal design performs comparably with the entropy criterion, and D- and A-optimality perform worse.



(a) *Comparison with A-, D- and E-optimality on temperature data*

(b) *Comparison with A-, D- and E-optimality on precipitation data, 111 node subsample*

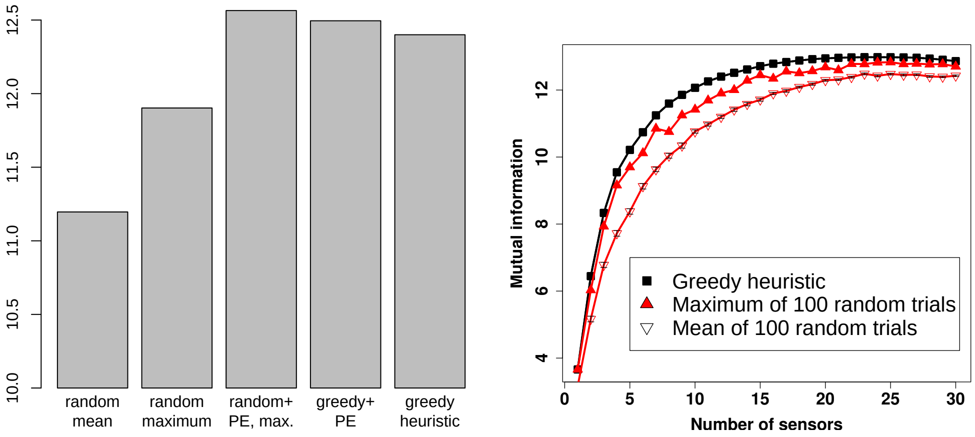
(c) *Comparison with D- and E-optimality on precipitation data*

(d) *Running time for 111 node precipitation subsample*

Figure 14: Comparison with classical experimental design. We plot RMS prediction error on 25% hold out target locations $\mathcal{U}$.

When we performed the same experiment with the precipitation data, SeDuMi ran out of memory (1 GB) for the SDP required to solve the A-optimality criterion. The largest subsample we could solve for all A-, D- and E-optimality on this data set was limited to 111 locations. Figure 14(b) presents the results. For the entire data set of 167 locations, we could still solve the D- and E-optimality SDPs. The results are presented in Figure 14(c). We can observe that for the 111 locations, D-optimality slightly outperforms mutual information. We have to consider, however, that the classical criteria are optimized to minimize the error covariance with respect to the locations $\mathcal{U}$ of interest, whereas mutual information merely tries to achieve uniformly low uncertainty over the entire space. For the full set of 167 locations, mutual information outperforms the other design criteria.

Figure 14(d) presents the running time for optimizing A-, D-, E-optimality, and mutual information, mutual information with truncation parameter $\varepsilon = 1$ and entropy on the 111 node subsample of the precipitation data on a Pentium M 1.7 GHz processor. We can see that optimizing entropy is fastest, closely followed by the truncated mutual information criterion described in Section 5.2 that is further evaluated in Section 9.7. Even without truncation, optimizing mutual information is three times faster than (fractionally) optimizing D-optimality and 24 times faster than A-optimality.



(a) *Objective values* (b) *Comparison of heuristics*

Figure 15: Comparison of the greedy algorithm with several heuristics.

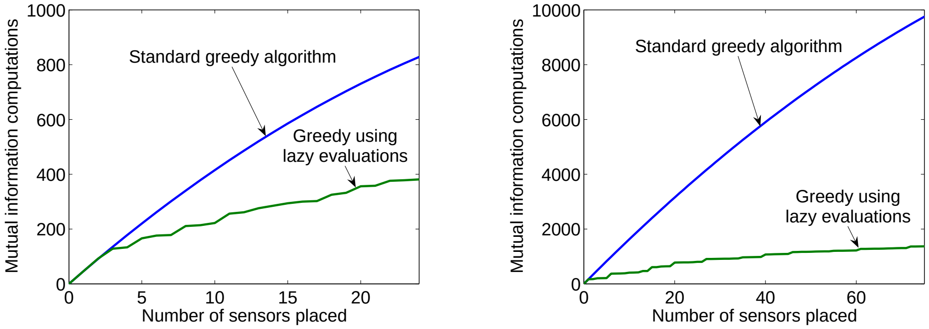
9.6 Empirical Analysis of the Greedy Algorithm

To study the effectiveness of the greedy algorithm, we compared the mutual information of the sets selected by our greedy algorithm to random selections, to a hill climbing method that uses a pairwise exchange heuristic, and—for small subsamples—to the bounds proved by the MIP as discussed in Section 4.5.

In this experiment, we used the empirical covariance matrix as the input to the algorithms. Figure 15(b) shows that the greedy algorithm provided significantly better results than the random selections, and even the maximum of a hundred random placements did not reach the quality of the greedy placements. Furthermore, we enhanced the random and greedy selections with the pairwise exchange (PE) heuristic, which iteratively finds exchanges of elements $y \in \mathcal{A}$ and $y' \in \mathcal{S} \setminus \mathcal{A}$ such that exchanging y and y' improves the mutual information score. Figure 15(a) presents objective values of these enhanced selection methods for a subset size of 12, for which the maximum over 100 random selections enhanced with PE actually exceeded the greedy score (unlike with most other subset sizes, where random + PE did about as well as the greedy algorithm). Typically, the objective values of random + PE, greedy + PE and greedy did not differ much. Note that as mentioned in Section 4, the performance guarantee for the greedy algorithm always provides an online approximation guarantee for the other heuristics.

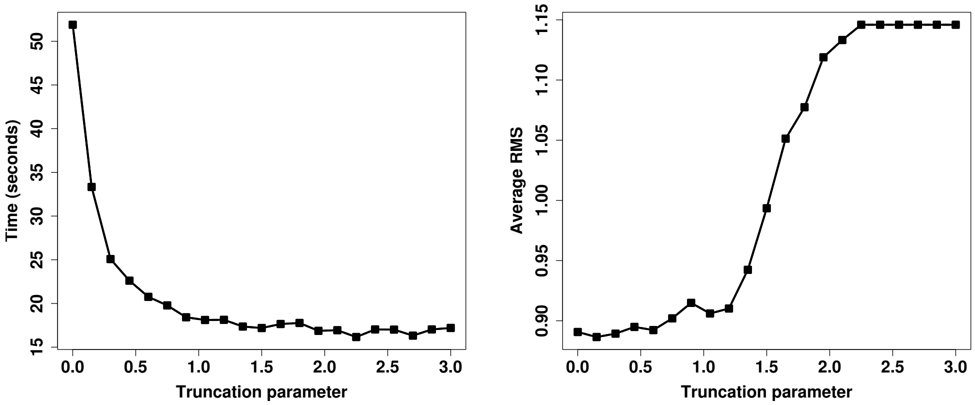
For a 16 node subsample of the temperature data set, we used the MIP from Section 4.5 to compute bounds on the optimal mutual information. Figure 5 presents the results. It can be seen, that for this small subsample, the greedy solution is never more than 5 percent away from the optimal solution, which is a much tighter bound than the a priori approximation factor of $(1 - 1/e)$.

We also experimented with the lazy evaluation strategy discussed in Section 5.1. For example when picking placements of size 50 for the precipitation data set, the number of mutual information computations decreased from 7125 to 1172, and the computation time on a Pentium M 1.7 GHz processor decreased from 41.3 seconds to 8.7 seconds. The results for both temperature and precipitation data sets are presented in Figures 16(a) and 16(b).



(a) *Temperature data* (b) *Precipitation data*

Figure 16: Performance improvements by using lazy evaluations of mutual information.



(a) *Running time* (b) *Approximation error*

Figure 17: Analysis of the experiments with local kernels. (a) running times for increasing level of truncation. (b) increase of average RMS error with increasing level of truncation. Note that for a truncation between 0.5 and 1.2, a good tradeoff between running time and error is achieved.

9.7 Results on Local Kernels

We also performed experiments to assess the running time versus quality trade-off incurred by using approximate local kernels. To provide intuition about the covariance structure, note that the 25, 50 and 75 percentiles of the absolute covariance entries were 0.122, 0.263 and 0.442, the maximum was 3.51, the minimum was $8.78E-6$. For the variance (the diagonal entries), the median was 1.70, and the minimum was 0.990. Figure 17(a) shows that the computation time can be drastically decreased as we increase the truncation parameter ε from 0 to the maximum variance. Figure 17(b) shows the RMS prediction accuracy for the 20 element subsets selected by Algorithm 3. According to the graphs, the range $\varepsilon \in [0.5, 1]$ seems to provide the appropriate trade-off between computation time and prediction accuracy.

In order to study the effect of local kernels on the placements, we performed the following experiment. We created a regular 7 by 7 grid with unit distance between neighboring grid points, and generated covariance matrices using two different GPs, one using the Gaussian (squared exponential) kernel, and the other using the local kernel (Equation 9). We exponentially increased the bandwidth in eight steps from 0.1 to 12.8. Figures 18 and 19 show the corresponding placements using mutual information to select the locations. From this experiment, we can see that the placements obtained using the non-local Gaussian kernel tend to be spread out slightly more, as one might expect. Overall, however, the placements appear to be very similar. In light of the computational advantages provided by local kernels, these results provide further evidence in the spirit of Section 9.7, namely that local kernels can be a valuable tool for developing efficient model-based sensor placement algorithms.

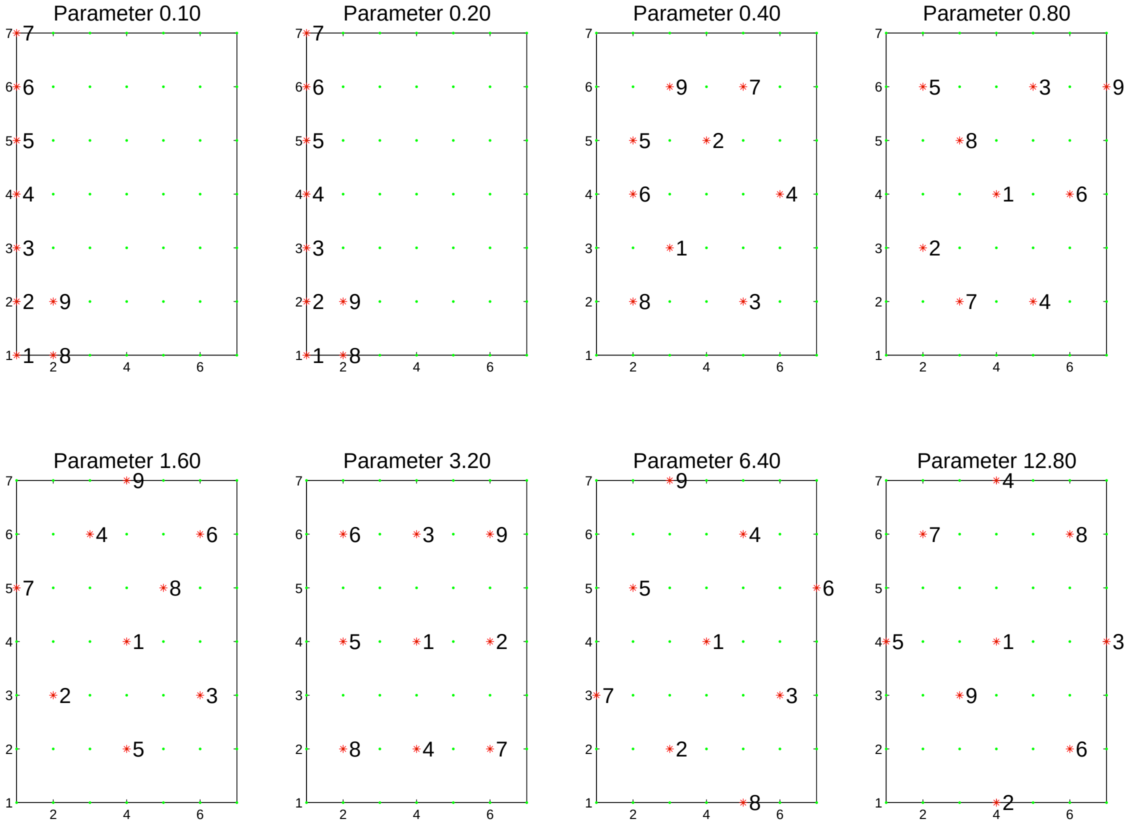


Figure 18: Placements under Gaussian kernel, mutual information criterion, increasing bandwidth

10. Future Work

There are several interesting possible extensions to the present work. Since the predictive variance in (2) does not depend on the actual observations, any closed-loop strategy which sequentially decides on the next location to measure, surprisingly, is equivalent to an open loop placement strategy which selects locations to make observations independently of the measured values. If there is uncertainty about the model parameters however, such as about the kernel bandwidths, then this is no longer true. In this case, we expect a sequential, closed-loop strategy to be more effective for predicting spatial phenomena. Krause and Guestrin (2007) present bounds comparing the performance of the optimal sequential strategy with the optimal fixed placement. This bound essentially depends on the

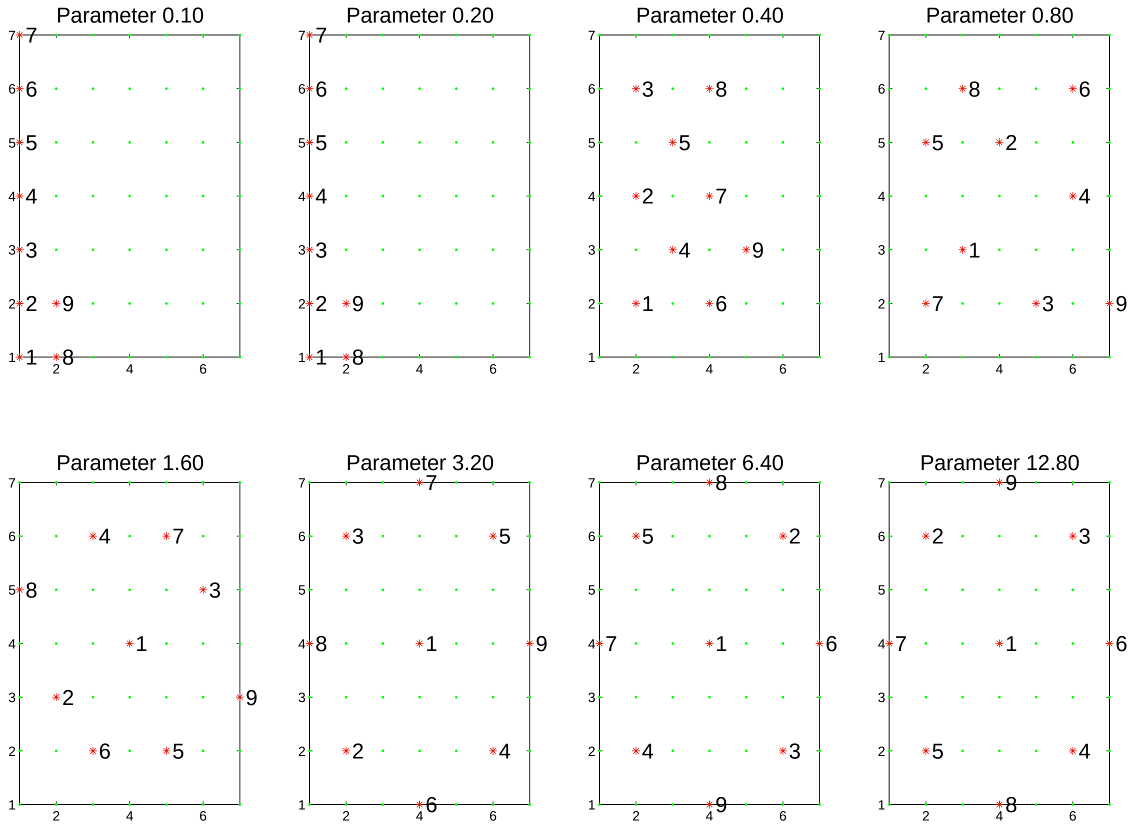


Figure 19: Placements under local kernel, mutual information criterion, increasing bandwidth

parameter entropy. We consider several exploration strategies for effectively reducing this parameter entropy and present sample complexity bounds. However, more work is needed in this area.

Another interesting open question is whether an approximation algorithm can be found for optimizing sensor placements subject to *submodular* cost functions—usually, the more sensors we have to buy, the cheaper they become per unit. To address this problem, Narasimhan and Bilmes (2006) present a submodular-supermodular procedure for bicriteria-optimization of a submodular function of which a submodular cost is subtracted. This procedure, while elegant, unfortunately can not provide approximation guarantees for this problem.

Of further interest are also constrained sensor placement problems, in which, for example, the placed sensors have to be connected in a routing tree, or have to lie on a collection of paths. Krause et al. (2006) provide evidence that submodularity can be leveraged to derive approximation algorithms for sensor placement even in these combinatorially even more challenging constrained optimization problems. However, there are still many open issues subject to further research.

11. Conclusions

In this paper, we tackle the problem of maximizing mutual information in order to optimize sensor placements. We prove that the exact optimization of mutual information is NP-complete, and provide an approximation algorithm that is within $(1-1/e)$ of the maximum mutual information configuration by exploiting the submodularity in the criterion. We also illustrate that submodularity can be used to obtain online bounds, which are useful for bounding the quality of the solutions obtained by any optimization method, and for designing branch and bound algorithms for the mutual information criterion. In order to scale up the application of our approach, show how to exploit lazy evaluations and local structure in GPs to provide significant speed-ups. We also extend our submodularity-based analysis of mutual information to incorporate robustness to sensor failures and model uncertainty.

Our very extensive empirical results indicate that data-driven placements can significantly improve the prediction accuracy over geometric models. We find, in contrast to previous work (Caselton et al., 1992; Zidek et al., 2000), that the mutual information criterion is often better than entropy and other classical experimental design criteria, both qualitatively and in prediction accuracy. In addition, the results show that a simple greedy algorithm for optimizing mutual information provides performance that is very close to the optimal solution in problems that are small enough to be solved exactly, and comparable to more complex heuristics in large problems.

We believe this work can be used to increase the efficacy of monitoring systems, and is a step towards well-founded active learning algorithms for spatial and structured data.

Acknowledgments

We would like to thank the Action Editor (Chris Williams) and the anonymous referees for their valuable comments and suggestions on this work. This work was partially supported by NSF Grant Nos. CNS-0509383, CNS-0625518 and a gift from Intel Corporation. Carlos Guestrin was partly supported by an Alfred P. Sloan Fellowship and an IBM Faculty Fellowship. Andreas Krause was partly supported by a Microsoft Research Graduate Fellowship. Ajit Singh acknowledges the support of NSERC.

Appendix A. Proofs

Proof [Theorem 2] Our reduction builds on the proof by Ko et al. (1995), who show that for any graph G, there exists a polynomially related, symmetric positive-definite matrix Σ such that Σ has a subdeterminant (of a submatrix resulting from the selection of k rows and columns $i_1,\ldots,i_k$) greater than some M if G has a clique of size at least k, and Σ does not have a subdeterminant greater than $M-\varepsilon$ for some (polynomially-large) $\varepsilon > 0$ if G does not have such a clique. Let G be a graph, and let Σ be the matrix constructed in Ko et al. (1995). We will consider Σ as the covariance matrix of a multivariate Gaussian distribution with variables $\mathcal{X}_\mathcal{U} = \{\mathcal{X}_1,\ldots,\mathcal{X}_n\}$. Introduce additional variables $\mathcal{X}_\mathcal{S} = \{y_1,\ldots,y_n\}$ such that $y_i|\mathcal{X}_i = x \sim \mathcal{N}(x,\sigma^2)$. Note that a subset

$\mathcal{A} \subseteq \mathcal{S}$, $|\mathcal{A}| = k$, has maximum entropy of all such subsets if and only if the parents $\Gamma_{\mathcal{A}} \subset \mathcal{U}$ of $\mathcal{A}$ have maximum entropy among all such subsets of $\mathcal{U}$. Now note that $I(\mathcal{A};(\mathcal{U}\cup\mathcal{S})\setminus\mathcal{A}) = H(\mathcal{A}) - H(\mathcal{A} \mid (\mathcal{U}\cup\mathcal{S})\setminus\mathcal{A}) = H(\mathcal{A}) - H(\mathcal{A} \mid \mathcal{U})$, because y_i and y_j are conditionally independent given $\mathcal{U}$. Furthermore, again because of independence, $H(\mathcal{A} \mid \mathcal{U})$ is a constant only depending on the cardinality of $\mathcal{A}$. Assume we could decide efficiently whether there is a subset $\mathcal{A} \subset \mathcal{S}$ such that $I(\mathcal{A};\mathcal{V}\setminus\mathcal{A}) \geq M'$. If we choose σ^2 small enough, then this would allow us to decide whether G has a clique of size k, utilizing the gap ε. ■

Proof [Lemma 5] Define $\hat{\mathcal{K}}(x,y) = \mathcal{K}(x,y)$ for $x \neq y$ and $\hat{\mathcal{K}}(x,x) = \mathcal{K}(x,x) + \sigma^2$ to include the sensor noise σ^2. Since $\mathcal{C}$ is compact and $\mathcal{K}$ continuous, $\mathcal{K}$ is uniformly continuous over $\mathcal{C}$. Hence, for any ε_1, there exists a δ_1 such that for all x,x',y,y', $\|x-x'\|_2 \leq \delta_1, \|y-y'\|_2 \leq \delta_1$ it holds that $|\,\mathcal{K}(x,y) - \mathcal{K}(x',y')| \leq \varepsilon_1$. Assume $\mathcal{C}_1 \subset \mathcal{C}$ is a finite mesh grid with mesh width $2\delta_1$. We allow sensor placement only on grid $\mathcal{C}_1$. Let $\mathcal{C}_2 \subset \mathcal{C}$ be a mesh grid of mesh width $2\delta_1$, which is derived by translating $\mathcal{C}_1$ by δ_1 in Euclidean norm, and let G_1, G_2 denote the restriction of the GP G to $\mathcal{C}_1, \mathcal{C}_2$. We assume $\mathcal{C}_1, \mathcal{C}_2$ cover $\mathcal{C}$ in the sense of compactness. We use the notation $\tilde{\cdot}$ to refer to the translated version in G_2 of the random variable $\cdot$ in G_1. $\hat{\mathcal{K}}$ is a symmetric strictly positive definite covariance function and $|\,\hat{\mathcal{K}}(X,y) - \hat{\mathcal{K}}(\tilde{X},\tilde{y})| \leq \varepsilon_1$ for all $X,y \in G_1$. Moreover, since $\mathcal{K}$ is positive semidefinite, the smallest eigenvalue of any covariance matrix derived from $\hat{\mathcal{K}}$ is at least σ^2.

Let $\mathcal{A}$ be a subset of $\mathcal{C}_1$ and $X \in \mathcal{C}_1 \setminus \mathcal{A}$. Using (5), we first consider the conditional variance $\sigma^2_{X|\mathcal{A}}$. By definition, $\|y-\tilde{y}\|_2 \leq \delta_1$, and hence $|\,\hat{\mathcal{K}}(X,y) - \hat{\mathcal{K}}(X,\tilde{y})| \leq \varepsilon_1$ for all $y \in \mathcal{A}$. Hence we know that $\|\Sigma_{\mathcal{A}\mathcal{A}} - \Sigma_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}}\|_2 \leq \|\Sigma_{\mathcal{A}\mathcal{A}} - \Sigma_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}}\|_F \leq k^2\varepsilon_1$. We furthermore note that $\|\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\|_2 = \lambda^{max}(\Sigma^{-1}_{\mathcal{A}\mathcal{A}}) = \lambda^{min}(\Sigma_{\mathcal{A}\mathcal{A}})^{-1} \leq \sigma^{-2}$, and hence

$$\begin{aligned}
\|\Sigma^{-1}_{\mathcal{A}\mathcal{A}} - \Sigma^{-1}_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}}\|_2 &= \|\Sigma^{-1}_{\mathcal{A}\mathcal{A}}(\Sigma_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}} - \Sigma_{\mathcal{A}\mathcal{A}})\Sigma^{-1}_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}}\|_2 \\
&\leq \|\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\|_2 \|\Sigma_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}} - \Sigma_{\mathcal{A}\mathcal{A}}\|_2 \|\Sigma^{-1}_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}}\|_2 \leq \sigma^{-4}k^2\varepsilon_1.
\end{aligned}$$

We derive $\|\Sigma_{X\tilde{\mathcal{A}}} - \Sigma_{X\mathcal{A}}\|_2 \leq \|\varepsilon_1 \mathbf{1}^T\|_2 = \varepsilon_1\sqrt{k}$, hence

$$\begin{aligned}
|\sigma^2_{X|A} - \sigma^2_{X|\tilde{\mathcal{A}}}| &= |\Sigma_{X\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\Sigma_{\mathcal{A}X} - \Sigma_{X\tilde{\mathcal{A}}}\Sigma^{-1}_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}}\Sigma_{\tilde{\mathcal{A}}X}| \\
&\leq\ 2\|\Sigma_{X\mathcal{A}} - \Sigma_{X\tilde{\mathcal{A}}}\|_2 \|\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\|_2 \|\Sigma_{X\mathcal{A}}\|_2 + \|\Sigma^{-1}_{\mathcal{A}\mathcal{A}} - \Sigma^{-1}_{\tilde{\mathcal{A}}\tilde{\mathcal{A}}}\|_2 \|\Sigma_{X\mathcal{A}}\|_2^2 + O(\varepsilon_1^2) \\
&\leq 2\varepsilon_1\sqrt{k}\sigma^{-2}M\sqrt{k} + \sigma^{-4}k^2\varepsilon_1 M^2 k + O(\varepsilon_1^2) \\
&\leq \varepsilon_1 k\sigma^{-2}M\left(2 + \sigma^{-2}k^2M\right) + O(\varepsilon_1^2),
\end{aligned}$$

where $M = \max_{x\in C}\mathcal{K}(x,x)$. We choose δ such that the above difference is bounded by $\sigma^2\varepsilon$. We note that (assuming w.l.o.g. $H(X \mid \mathcal{A}) \geq H(X \mid \tilde{\mathcal{A}})$)

$$H(X \mid \mathcal{A}) - H(X \mid \tilde{\mathcal{A}}) = \frac{1}{2}\log\frac{\sigma^2_{X|\mathcal{A}}}{\sigma^2_{X|\tilde{\mathcal{A}}}} \leq \frac{\log(1+\varepsilon)}{2} \leq \frac{\varepsilon}{2}.$$

which concludes the argument. ■

Proof [Corollary 6] The higher order terms $O(\varepsilon_1^2)$ can be worked out as $k\sigma^{-2}\varepsilon^2(1 + Mk^2\sigma^{-2} + \varepsilon k^2\sigma^{-2})$. Assuming that $\varepsilon < \min(M,1)$, this is bounded by $3k^3M\sigma^{-4}\varepsilon$. Using the Lipschitz assumption, we can directly compute δ_1 from ε_1 in the above proof, by letting $\delta = \varepsilon_1/L$. Let $R =$

$k\sigma^{-2}M\left(2+\sigma^{-2}k^2M\right)+3k^3M\sigma^{-4}$. We want to choose δ such that $\varepsilon_1 R \leq \sigma^2\varepsilon$. Hence if we choose $\delta \leq \frac{\sigma^2\varepsilon}{LR}$, then $|H(X \mid \mathcal{A}) - H(X \mid \tilde{\mathcal{A}})| \leq \varepsilon$ uniformly as required. Note that in order to apply the result from Nemhauser et al. (1978), the approximate monotonicity has to be guaranteed for subsets of size $2k$, which results in the stated bound. ■

Proof [Theorem 7] The following proof is an extension of the proof by Nemhauser et al. (1978), using some simplifications by Jon Kleinberg.

Let $s_1,\ldots,s_k$ be the locations selected by the greedy algorithm. Let $\mathcal{A}_i = \{s_1,\ldots,s_i\}$, $\mathcal{A}^*$ be the optimal solution, and $\delta_i = \mathrm{MI}(\mathcal{A}_i) - \mathrm{MI}(\mathcal{A}_{i-1})$. By *Lemma* 5, we have, for all $1 \leq i \leq k$,

$$\mathrm{MI}(\mathcal{A}_i \cup \mathcal{A}^*) \geq \mathrm{MI}(\mathcal{A}^*) - k\varepsilon.$$

We also have, for $0 \leq i < k$,

$$\mathrm{MI}(\mathcal{A}_i \cup \mathcal{A}^*) \leq \mathrm{MI}(\mathcal{A}_i) + k\delta_{i+1} = \sum_{j=1}^{i} \delta_j + k\delta_{i+1}.$$

Hence we have the following sequence of inequalities:

$$\begin{gathered}\mathrm{MI}(\mathcal{A}^*) - k\varepsilon \leq k\delta_1 \\ \mathrm{MI}(\mathcal{A}^*) - k\varepsilon \leq \delta_1 + k\delta_2 \\ \vdots \\ \mathrm{MI}(\mathcal{A}^*) - k\varepsilon \leq \sum_{j=1}^{k-1} \delta_j + k\delta_k.\end{gathered}$$

Now we multiply both sides of the i-th inequality by $\left(1-\frac{1}{k}\right)^{k-1}$, and add all inequalities up. After cancellation, we get

$$\left(\sum_{i=0}^{k-1} (1-1/k)^i\right)(\mathrm{MI}(\mathcal{A}^*) - k\varepsilon) \leq k\sum_{i=1}^{k} \delta_i = k\,\mathrm{MI}(\mathcal{A}_k).$$

Hence, as claimed, with $\mathcal{A}_G = \mathcal{A}_k$ (i.e., $\mathcal{A}_G$ is the k-element greedy solution)

$$\mathrm{MI}(\mathcal{A}_G) \geq \left(1-(1-1/k)^k\right)(\mathrm{MI}(\mathcal{A}^*) - k\varepsilon) \geq (1-1/e)(\mathrm{MI}(\mathcal{A}^*) - k\varepsilon).$$

■

Proof [Remark 12] We have that $H(y \mid Z) < 0 \Leftrightarrow \mathcal{K}(y,y) + \sigma^2 - \frac{\mathcal{K}(Z,y)^2}{\mathcal{K}(Z,Z)+\sigma^2} < \frac{1}{2\pi e}$. Using a similar argument as the proof of Lemma 5, for very fine discretizations, there exists a y arbitrarily close to Z, such that for any $\alpha > 0$, $|\mathcal{K}(Z,Z) - \mathcal{K}(y,y)| \leq \alpha$ and $|\mathcal{K}(Z,Z) - \mathcal{K}(Z,y)| \leq \alpha$. Plugging these bounds into the definition of $H(y \mid Z)$ and some algebraic manipulation proves the claim. ■

References

A. C. Atkinson. Recent developments in the methods of optimum and related experimental designs. *International Statistical Review / Revue Internationale de Statistique*, 56(2):99–115, Aug. 1988.

A. C. Atkinson. The usefulness of optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):59–76, 1996.

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.

X. Bai, S. Kumar, Z. Yun, D. Xuan, and T. H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Florence, Italy, 2006.

J. M. Bernardo. Expected information as expected utility. *Annals of Statistics*, 7(3):686–690, May 1979.

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

W. F. Caselton and T. Hussain. Hydrologic networks: Information transmission. *Journal of Water Resources Planning and Management*, WR2:503–520, 1980.

W. F. Caselton and J. V. Zidek. Optimal monitoring network designs. *Statistics and Probability Letters*, 2(4):223–227, 1984.

W. F. Caselton, L. Kan, and J. V. Zidek. *Statistics in the Environmental and Earth Sciences*, chapter Quality data networks that minimize entropy, pages 10–38. Halsted Press, 1992.

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

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.

R. D. Cook and C. J. Nachtsheim. A comparison of algorithms for constructing exact D-optimal designs. *Technometrics*, 22(3):315–324, Aug. 1980. ISSN 00401706.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Interscience, 1991.

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

C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991.

A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *Symposium on the Theory of Computing*, 2008.

A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB)*, 2004.

V. Federov and W. Mueller. Comparison of two approaches in the optimal design of an observation network. *Statistics*, 20:339–351, 1989.

V. V. Fedorov. *Theory of Optimal Experiments*. Academic Press, 1972. Trans. W. J. Studden and E. M. Klimko.

P. Flaherty, M. Jordan, and A. Arkin. Robust design of biological experiments. In *Advances in Neural Information Processing Systems (NIPS) 19*, 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.

G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins, 1989.

H. H. Gonzalez-Banos and J. Latombe. A randomized art-gallery algorithm for sensor placement. In *Proc. 17th ACM Symposium on Computational Geometry*, pages 232–240, 2001.

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.

P. Guttorp, N. D. Le, P. D. Sampson, and J. V. Zidek. Using entropy in the redesign of an environmental monitoring network. Technical report, Department of Statistics. University of British Columbia., 1992. Tech. Rep. 116.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2003.

N. Heo and P. K. Varshney. Energy-efficient deployment of intelligent mobile sensor networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35(1):78–92, 2005.

D. S. Hochbaum and W. Maas. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32:130–136, 1985.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

A. Howard, M. Mataric, and G. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem, 2002.

R. Kershner. The number of circles covering a set. *American Journal of Mathematics*, 61:665–671, 1939.

C. Ko, J. Lee, and M. Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

A. Krause and C. Guestrin. A note on the budgeted maximization of submodular functions. Technical report, CMU-CALD-05-103, 2005.

A. Krause and C. Guestrin. Nonmyopic active learning of gaussian processes: An exploration–exploitation approach. In *International Conference on Machine Learning*, 2007.

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.

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.

U. Lerner and R. Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI)*, 2001.

D. V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27:986–1005, 1956.

D. V. Lindley and A. F. M. Smith. Bayes' estimates for the linear model. *Journal of the Royal Statistical Society, Ser. B*, 34:1–18, 1972.

S. P. Luttrell. The use of transinformation in the design of data sampling schemes for inverse problems. *Inverse Problems*, 1:199–218, 1985.

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

D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge UP, 2003.

G. P. McCabe. Principal variables. *Technometrics*, 26(2):137–144, 1984.

M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2): 239–245, May 1979.

R. K. Meyer and C. J. Nachtsheim. Constructing exact D-optimal experimental designs by simulated annealing. *American Journal of Mathematical and Management Sciences*, 8(3-4):329–359, 1988.

T. J. Mitchell. An algorithm for the construction of ”D-optimal” experimental designs. *Technometrics*, 16(2):203–210, May 1974a. ISSN 00401706.

T.J. Mitchell. Computer construction of ”D-optimal” first-order designs. *Technometrics*, 16(2): 211–220, May 1974b. ISSN 00401706.

Avidan Moghaddam, Weiss. Fast pixel/part selection with sparse eigenvectors. In *International Conference on Computer Vision*, 2007.

B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact and greedy algorithms. In *Advances in Neural Information Processing Systems (NIPS) 18*, 2005.

B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse LDA. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML)*, 2006.

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.

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

G. L. Nemhauser and L. A. Wolsey. *Studies on Graphs and Discrete Programming*, chapter Maximizing submodular set functions: Formulations and analysis of algorithms, pages 279–301. North-Holland, 1981.

N-.K Nguyen and A. J. Miller. A review of some exchange algorithms for constructing discrete D-optimal designs. *Computational Statistics and Data Analysis*, 14:489–498, 1992.

D. J. Nott and W. T. M. Dunsmuir. Estimation of nonstationary spatial covariance structure. *Biometrika*, 89:819–829, 2002.

A. O'Hagan. Curve fitting and optimal design for prediction (with discussion). *Journal of the Royal Statistical Society, Ser. B*, 40:1–42, 1978.

C. J. Paciorek. *Nonstationary Gaussian Processes for Regression and Spatial Modelling*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, May 2003.

L. Paninski. Asymptotic theory of information-theoretic experimental design. *Neural Computation*, 17(7):1480–1507, 2005.

F. Pukelsheim. Information increasing orderings in experimental design theory. *International Statistical Review / Revue Internationale de Statistique*, 55(2):203–219, Aug. 1987. ISSN 03067734.

N. Ramakrishnan, C. Bailey-Kellogg, S. Tadepalli, and V. N. Pandey. Gaussian processes for active data mining of spatial aggregates. In *SIAM Data Mining*, 2005.

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

P. J. Ribeiro Jr. and P. J. Diggle. geoR: A package for geostatistical analysis. R-NEWS Vol 1, No 2. ISSN 1609-3631, 2001.

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.

J. Sacks, S. B. Schiller, and W. J. Welch. Designs for computer experiments. *Technometrics*, 31(1): 41–47, Feb. 1989. ISSN 00401706.

P. Sebastiani and H. P. Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society, Series B*, 62(1):145–157, 2000.

M. Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14 (2):69–106, 2004.

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.

M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14: 165–170, 1987.

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

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NIPS) 18*, 2005.

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

A. J. Storkey. Truncated covariance matrices and Toeplitz methods in Gaussian processes. In *Artificial Neural Networks - ICANN 1999*, 1999.

J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software, special issue on interior-point methods*, 11(12):625–653, 1999.

M. Sviridenko. A note on maximizing a submodular set function subject to knapsack constraint. *Operations Research Letters*, 32:41–43, 2004.

S. Toumpis and G. A. Gupta. Optimal placement of nodes in large sensor networks under a general physical layer model. In *Proc. IEEE Communications Society Conference on Sensor and Ad Hoc Communications (SECON)*, 2005.

W. J. Welch. Branch-and-bound search for experimental design based on D-optimality and other criteria. *Technometrics*, 24(1):41–48, 1982.

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.

S. Wu and J. V. Zidek. An entropy based review of selected NADP/NTN network sites for 1983–86. *Atmospheric Environment*, 26A:2089–2103, 1992.

D. Ylvisaker. *A Survey of Statistical Design and Linear Models*, chapter Design on random fields. North-Holland, 1975.

K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML)*, 2006.

Z. Zhu and M. L. Stein. Spatial sampling design for prediction with estimated parameters. *Journal of Agricultural, Biological and Environmental Statistics*, 11:24–49, 2006.

J. V. Zidek, W. Sun, and N. D. Le. Designing and integrating composite networks for monitoring multivariate gaussian pollution fields. *Applied Statistics*, 49:63–79, 2000.

D. L. Zimmerman. Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction. *Environmetrics*, 17(6):635–652, 2006.