Near-Optimal Sensor Placements in Gaussian Processes

Carlos Guestrin GUESTRIN@CS.CMU.EDU
Andreas Krause KRAUSEA@CS.CMU.EDU
Ajit Paul Singh AJIT@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University

Abstract

When monitoring spatial phenomena, which are often modeled as Gaussian Processes (GPs), choosing sensor locations is a fundamental task. A common strategy is to place sensors at the points of highest entropy (variance) in the GP model. We propose a *mutual information* criteria, and show that it produces better placements. Furthermore, we prove that 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 our criterion. This algorithm is extended to handle local structure in the GP, yielding significant speedups. We demonstrate the advantages of our approach on two real-world data sets.

1. Introduction

When monitoring spatial phenomena, such as temperatures in an indoor environment as shown in Fig. 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 (*c.f.*, (Hochbaum & Maas, 1985; Gonzalez-Banos & Latombe, 2001)). In practice, however, this 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 Fig. 1(b). Furthermore, note that correlations can be both positive and negative, as shown in Fig. 1(c). Often, correlations may be too weak to enable prediction from a single sensor, suggesting the need for combining data from multiple sensors to obtain accurate predictions.

An alternative approach from spatial statistics (Cressie, 1991), making weaker assumptions, 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 to represent uncertainty about the sensed field. The learned GP model can then be used to predict the effect of placing sensors at particular locations, and thus optimize their positions. 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).

Typical sensor placement techniques greedily add sensors where uncertainty about the phenomena is highest, i.e., the highest entropy location of the GP (Cressie, 1991). 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, i.e., 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), e.g., Fig. 2(c). Since a sensor usually provides information about the area around it, a sensor on the boundary “wastes” sensed information.

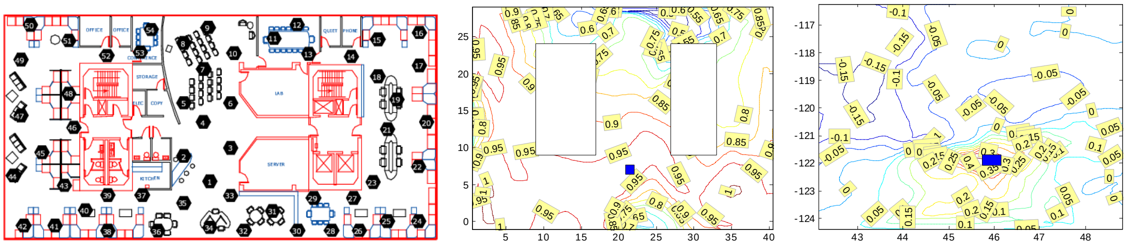
In this paper, we address this issue by proposing a new optimization criterion, *mutual information*, that seeks to find sensor placements that are most informative about unsensed locations. Our optimization criterion *directly* measures the effect of sensor placements on the posterior uncertainty of the GP. We first prove that maximizing mutual information is an NP-complete problem. Then, by exploiting the fact that mutual information is a *submodular* function (Nemhauser et al., 1978), we design an 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 other GP-based sensor placement approaches. Though polynomial, the complexity of our basic algorithm is relatively high – $\mathcal{O}(kn^4)$ to select k out of n possible sensor locations. If we trim low covariance entries, exploiting locality in sensing areas, we reduce the complexity to $\mathcal{O}(kn)$.

2. Gaussian processes

Consider, for example, a sensor network, such as the one we deployed as shown in Fig. 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 (Deshpande et al., 2004), approach is to assume that the temperatures have a (multivariate) Gaussian joint distribution. Here, we have a set of n random variables X with joint distribution:

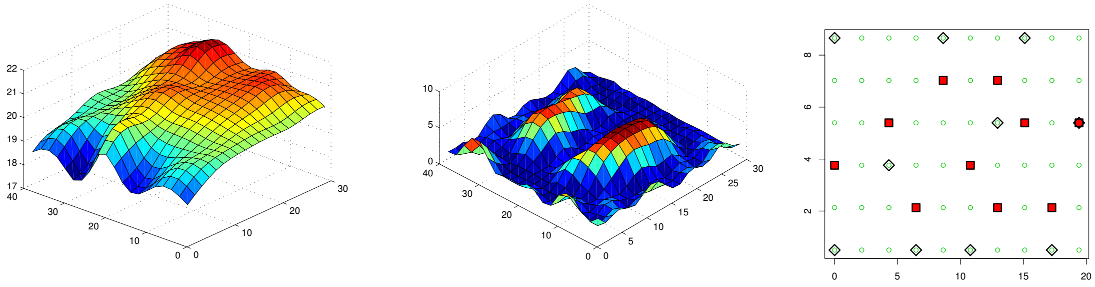
$$P(X = x) = \frac{1}{(2\pi)^{n/2}|\Sigma|} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)},$$

Appearing in *Proceedings of the 22^{nd} International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).



(a) *54 node sensor network deployment* (b) *Temperature correlations* (c) *Precipitation correlations*

Figure 1. Correlation between a sensor placed on the blue square and other possible locations for: (b) temperature data from the sensor network deployment in Fig. 1(a); (c) precipitation data from measurements made across the Pacific Northwest, Fig. 5(a).



(a) *Temperature prediction using GP* (b) *Variance of temperature prediction* (c) *Entropy vs. MI placements*

Figure 2. Using all sensors (a) Predicted temperature (b) predicted variance. An example of placements chosen using entropy and mutual information in (c). Diamonds indicate the positions chosen using entropy; squares the positions chosen using MI.

where μ is the mean vector and Σ is the covariance matrix. If we consider indexing each variable $X_i \in X$ by i, then we will have a finite set of indices $\mathcal{V}$, in our sensor network example $|\mathcal{V}| = 54$. Interestingly, if we consider any subset of our random variables, $\mathcal{A} \subseteq \mathcal{V}$, then their joint distribution will also be Gaussian.

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 such prediction (Golub & Van Loan, 1989). Although linear regression often gives excellent predictions, there is usually no notion of uncertainty, e.g., for Fig. 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 consider an infinite number of random variables. In analogy to the multivariate Gaussian above where the the index set $\mathcal{V}$ was finite, we now have an (possibly uncountably) infinite index set $\mathcal{V}$. In our temperature example, $\mathcal{V}$ would be a subset of $\mathbb{R}^2$, and each index would correspond to a position in the lab. GPs have been widely studied, *c.f.*, (MacKay, 2003; Paciorek, 2003; Seeger, 2004).

An important property of GPs is that for every finite subset $\mathcal{A}$ of the indices $\mathcal{V}$, the joint distribution over these random variables is Gaussian, e.g., 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. In this paper, we *do not* make other limiting assumptions, such as $\mathcal{K}(\cdot, \cdot)$ being stationary or isotropic. 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 $\mathcal{A}$ 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 $x_{\mathcal{A}}$ corresponding to the finite subset $\mathcal{A} \subset \mathcal{V}$, we can predict the value at every point $y \in \mathcal{V}$ conditioned on these measurements, $P(X_y \mid x_{\mathcal{A}})$. The distribution of 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:

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

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

where $\Sigma_{y\mathcal{A}}$ is a covariance vector with one entry for each $u \in \mathcal{A}$ with value $\mathcal{K}(y, u)$. Figures Fig. 2(a) and Fig. 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 are 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 Fig. 2(b) will not depend on the observed temperatures.

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 is called *sampling design*: finding the k best sensor locations out of a finite subset $\mathcal{V}$ of possible locations, e.g., out of a grid discretization of $\mathbb{R}^2$.

An often used heuristic is to start from an empty set of locations, $\mathcal{A} = \emptyset$, and greedily add placements until $|\mathcal{A}| = k$. Usually, at each iteration, the greedy rule used is to add the location $y \in \mathcal{V} \setminus \mathcal{A}$ that has highest variance according to Eq. (2) (McKay et al., 1979; Cressie, 1991), i.e., the location that we are most uncertain about given the sensors placed thus far. Since, for a fixed kernel function, the variance does not depend on the observed values, this optimization can be done before deploying the sensors.

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

$$H(Y|\mathcal{A}) = \frac{1}{2}\log(2\pi e \sigma^2_{Y|\mathcal{A}}), \tag{3}$$

where, with some abuse of notation, we use $\mathcal{A}$ to refer to a set of indices and the corresponding set of random variables. If we define the set of selected locations as $\mathcal{A} = \{Y_1, \ldots, Y_k\}$, using the chain-rule of entropies, we have that:

$$H(\mathcal{A}) = H(Y_k \mid Y_1, \ldots, Y_{k-1}) + ... + H(Y_2 \mid Y_1) + H(Y_1).$$

Thus, we can view the greedy variance heuristic as an approximation to the problem

$$\operatorname*{argmax}_{\mathcal{A}:|\mathcal{A}|=k} H(\mathcal{A}), \tag{4}$$

that is, finding the set of sensor locations that has maximum joint entropy. This is an intuitive criterion 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 Fig. 2(c), where sensors are placed far apart along the boundary of the space and information is "wasted". 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*: rather than considering prediction quality over the space of interest, the entropy criterion only considers the entropy of the selected sensor locations. In this paper, we propose a new optimization criterion that addresses this problem: we search for the subset of sensor locations that most significantly reduces the uncertainty about the estimates in the rest of the space.

More formally, we define 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 an additional 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 throughout $\mathcal{V}$. Specifically,

$$\operatorname*{argmax}_{\mathcal{A} \subseteq \mathcal{S}:|\mathcal{A}|=k} H(\mathcal{V} \setminus \mathcal{A}) - H(\mathcal{V} \setminus \mathcal{A} \mid \mathcal{A}), \tag{5}$$

that is, the set $\mathcal{A}$ that maximally reduces the entropy over the rest of the space $\mathcal{V} \setminus \mathcal{A}$. Note that our criterion $H(\mathcal{V} \setminus \mathcal{A}) - H(\mathcal{V} \setminus \mathcal{A} \mid \mathcal{A})$ is equivalent to finding the set that maximizes the *mutual information* $I(\mathcal{A}; \mathcal{V} \setminus \mathcal{A})$ between $\mathcal{A}$ and the rest of the space $\mathcal{V} \setminus \mathcal{A}$. On the same simple example in Fig. 2(c), 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 Sec. 6 further demonstrate the advantages in performance of our mutual information criterion.

Entropy and mutual information are both hard to optimize. Maximizing either criteria is NP-complete.

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 $H(\mathcal{A}) \geq M$ is NP-complete.* □

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 $I(\mathcal{A}; \mathcal{V} \setminus \mathcal{A}) \geq M$ is NP-complete.* □

Proofs of all results are given in the appendix.

4. Approximation algorithm

Optimizing our mutual information criterion is an NP-complete problem. We now describe a poly-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 $F(\mathcal{A}) = I(\mathcal{A}; \mathcal{V} \setminus \mathcal{A})$, our goal is to greedily select the next sensor Y that maximizes:

$$\begin{aligned} F(\mathcal{A} \cup Y) - F(\mathcal{A}) = \\ &= H(\mathcal{A} \cup Y) - H(\mathcal{A} \cup Y|\bar{\mathcal{A}}) - \left[H(\mathcal{A}) - H(\mathcal{A}|\bar{\mathcal{A}} \cup Y)\right], \\ &= H(Y|\mathcal{A}) - H(Y|\bar{\mathcal{A}}), \end{aligned}$$

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 only considers the $H(Y|\mathcal{A})$ part. In contrast, our greedy mutual information trades off uncertainty with $-H(Y|\bar{\mathcal{A}})$, which forces us to pick a Y that is "central" with respect to the unselected locations $\bar{\mathcal{A}}$. Using the definition of conditional entropy in Eq. (3), 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 to the optimal sensor placement: If the algorithm returns set $\widehat{A}$, then

$$I(\widehat{A}; \mathcal{V} \setminus \widehat{A}) \geq (1 - 1/e) \max_{\mathcal{A} \subset \mathcal{S}, |\mathcal{A}|=k} I(\mathcal{A}; \mathcal{V} \setminus \mathcal{A}) - k\varepsilon,$$

```
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
        $Y^* \leftarrow \underset{\substack{Y \in \mathcal{S} \setminus \mathcal{A}: \\ \bar{\mathcal{A}} = \mathcal{V} \setminus (\mathcal{A} \cup Y)}}{\operatorname{argmax}} \frac{\sigma_Y^2 - \Sigma_{Y\mathcal{A}} \Sigma_{\mathcal{A}\mathcal{A}}^{-1} \Sigma_{\mathcal{A}Y}}{\sigma_Y^2 - \Sigma_{Y\bar{\mathcal{A}}} \Sigma_{\bar{\mathcal{A}}\bar{\mathcal{A}}}^{-1} \Sigma_{\bar{\mathcal{A}}Y}}$;
        $\mathcal{A} \leftarrow \mathcal{A} \cup Y^*$;
end
```

Algorithm 1: Approximation algorithm for maximizing mutual information.

for some small $\varepsilon > 0$.

To prove this result, we use *submodularity* (Nemhauser et al., 1978), a property of set functions that 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, $H(Y|\mathcal{A}) \geq H(Y|\mathcal{A} \cup \mathcal{B})$ (Cover & Thomas, 1991), note that our greedy update rule maximizing $H(Y|\mathcal{A}) - H(Y|\bar{\mathcal{A}})$ implies that

$$F(\mathcal{A}' \cup Y) - F(\mathcal{A}') \leq F(\mathcal{A} \cup Y) - F(\mathcal{A}),$$

when $\mathcal{A} \subseteq \mathcal{A}'$, i.e., adding Y to $\mathcal{A}$ helps more than adding Y to $\mathcal{A}'$. This is exactly the definition of a submodular function. Hence we have the following result:

Lemma 3. *The function $\mathcal{A} \mapsto I(\mathcal{A}; V \setminus \mathcal{A})$ is submodular.* □

A submodular set function F is called nondecreasing if $F(\mathcal{A} \cup Y) \geq F(\mathcal{A})$ for $Y \in \mathcal{V}$. In (Nemhauser et al., 1978), it is proven that for nondecreasing submodular set functions F with $F(\emptyset) = 0$, the greedy algorithm guarantees a performance guarantee of $(1-1/e)OPT$, where OPT is the value of the optimal subset of size k. This greedy algorithm is defined by selecting in each step the element $Y^* = \operatorname{argmax}_Y F(\mathcal{A} \cup Y) - F(\mathcal{A})$. This is exactly the algorithm we propose in the previous section.

It is clear that $F(\emptyset) = I(\emptyset; \mathcal{V}) = 0$, as required by Nemhauser et al. (1978). However, the monotonicity of our objective function is not apparent, since $F(\mathcal{V}) = I(\mathcal{V}, \emptyset) = 0$, our objective function will increase and then decrease, and, thus, is not monotonic. Fortunately, the proof of Nemhauser et al. (1978) does not use monotonicity for all possible sets, it is sufficient to prove that F is monotonic for all sets of size up to $2k$. Intuitively, mutual information is not monotonic when the set of sensor selected 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 Nemhauser et al. (1978).

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 non-decreasing* for sets of size up to $2k$. More precisely we prove the following result:

Lemma 4. *Let G be a Gaussian process on a compact subset $\mathcal{C}$ of $\mathbb{R}^m$ with a positive-definite, continuous covariance kernel $\mathcal{K} : \mathcal{C} \times \mathcal{C} \to \mathbb{R}_0^+$. 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 $\forall Y \in \mathcal{V} \setminus \mathcal{A}, F(\mathcal{A} \cup Y) - F(\mathcal{A}) \geq -\varepsilon$ for all $\mathcal{A} \subseteq \mathcal{S}$, $|\mathcal{A}| \leq 2k$.* □

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

Corollary 5. *If $\mathcal{K}$ is Lipschitz-continuous with constant L, then the required discretization is*

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

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

Corollary 5 guarantees that for any $\varepsilon > 0$, a polynomial discretization level is sufficient to guarantee that mutual information is $\varepsilon-$approximately non-decreasing. These bounds on the discretization are, of course, worst case bounds considering sensor placements that are arbitrarily close to each other. We expect the bounds to be very loose in the situations that arise during normal operation of the greedy algorithm, since the algorithm is unlikely to place sensors at such a close proximity.

Combining our Lemmas 3 and 4 with the Nemhauser et al. (1978) result, we obtain our constant-factor approximation bound on the quality of the sensor placements obtained by our algorithm:

Theorem 6. *Under the assumptions of Lemma 4, Algorithm 1 is guaranteed to select a set $\mathcal{A}$ of k sensors for which*

$$I(\mathcal{A}; V \setminus \mathcal{A}) \geq (1 - 1/e)(OPT - k\varepsilon),$$

where OPT is the value of 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.

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, and the task is to select placements whose total cost is within our budget. We have recently extended the submodular function maximization approach of Nemhauser et al. (1978) to address this budgeted case (Krause & Guestrin, 2005). The combination of the analysis in this paper with this new result also yields a constant-factor $(1 - 1/e)$ approximation guarantee for the sensor placement problem with non-uniform costs.

4.3. 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 for sensor placement. The corresponding greedy algorithm adds the sensor Y maximizing $H(\mathcal{A} \cup Y) - H(\mathcal{A}) = H(Y|\mathcal{A})$. Unfortunately, our analysis of approximate monotonicity does not extend to the entropy case: Consider $H(Y|\mathcal{A})$ for $\mathcal{A} = \{Z\}$, for sufficiently small measurement noise σ^2, we show that $H(Y|Z)$ can become arbitrarily negative as the mesh width of the discretization decreases. Thus, (even approximate) non-decreasingness 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 7. *Under the same assumptions as in Lemma 4, for any $\varepsilon > 0$, there exists a mesh discretization width $\delta > 0$ such that entropy violates the monotonicity criteria by at least ε, if $\sigma^2 < \frac{1}{4\pi e}$.* □

5. Local kernels

Greedy updates for both entropy and mutual information require the computation of conditional entropies using Eq. (3), which involves solving a system of $|\mathcal{A}|$ linear equations. For entropy maximization, where we consider $H(Y|\mathcal{A})$ alone, the complexity of this operation is $\mathcal{O}(k^3)$. To maximize the mutual information, we also need $H(Y|\bar{\mathcal{A}})$ requiring $\mathcal{O}(n^3)$, for $n = |\mathcal{V}|$. Since we need to recompute the score of all possible position at every iteration of Algorithm 1, the complexity of our greedy approach for selecting k sensors is $\mathcal{O}(kn^4)$, which is not computationally feasible for very fine discretizations (large n). We address this issue by exploiting locality in the kernel function: 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, i.e., that has non-zero value only for a small portion of the space (Storkey, 1999).

Even if the covariance function does not have compact support, it can be appropriate to compute $H(Y|\tilde{\mathcal{B}}) \approx H(Y|\mathcal{B})$ 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 $H(Y|\mathcal{B} \setminus X) - H(Y|\mathcal{B})$ is bounded by $\varepsilon^2/(\sigma^2\sigma^2_X)$ (using a similar argument as the proof of Lemma 4), assuming that each sensor has independent Gaussian measurement error of at least σ^2. The total decrease of entropy $H(Y|\tilde{\mathcal{B}}) - H(Y|\mathcal{B})$ is bounded by $n\varepsilon^2/\sigma^4$. This truncation allows to compute $H(Y|\bar{\mathcal{A}})$ much more efficiently, at the expense of this small absolute error. In the special case of isotropic covariance functions, 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 $H(Y|\bar{\mathcal{A}})$ from $\mathcal{O}(n^3)$ to $\mathcal{O}(d^3)$, which is a constant.

```
     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        $\delta_Y \leftarrow H(Y) - \tilde{H}_\varepsilon(Y|\mathcal{V} \setminus Y)$;
       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          $\delta_Y \leftarrow \tilde{H}_\varepsilon(Y|\mathcal{A}) - \tilde{H}_\varepsilon(Y|\bar{\mathcal{A}})$;
     end
```

Algorithm 2: Approximation algorithm for maximizing mutual information using local kernels.

Our truncation approach leads to the more efficient optimization algorithm shown in Algorithm 2. 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 $\mathcal{O}(nd^3)$ operations. For each one of the k iterations, finding the next sensor (Line 2) requires $\mathcal{O}(n)$ comparisons, and adding the new sensor Y^* can only change the score of its neighbors ($N(Y^*; \varepsilon) \leq d$), thus Line 3 requires $\mathcal{O}(d \cdot d^3)$ operations. The total running time of Algorithm 2 is $\mathcal{O}(nd^3 + kn + kd^4)$, which can be significantly lower than the $\mathcal{O}(kn^4)$ operations required by Algorithm 1. We summarize our analysis with the following theorem:

Theorem 8. *Under the assumptions of Lemma 4, guaranteeing ε_1-approximate non-decreasingness and truncation parameter ε_2, Algorithm 2 selects $\mathcal{A} \subseteq \mathcal{S}$ such that*

$$I(\mathcal{A}; V \setminus \mathcal{A}) \geq (1 - 1/e)(OPT - k\varepsilon_1 - 2kn\varepsilon_2/\sigma^4),$$

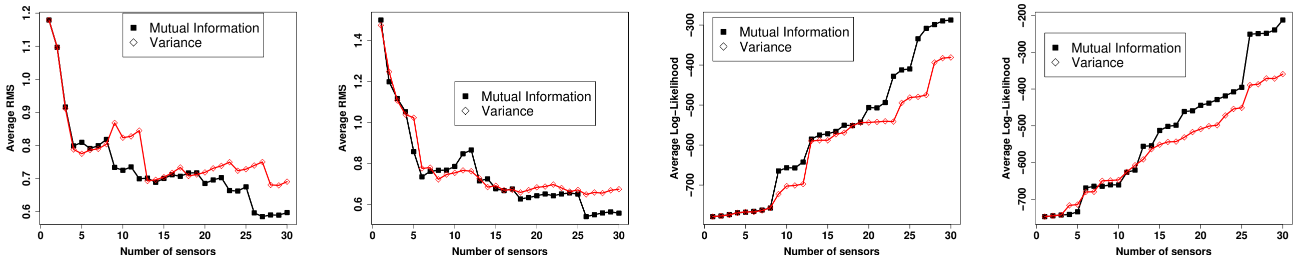
in time $\mathcal{O}(nd^3 + nk + kd^4)$. □

It is possible to slightly improve the performance of Algorithm 2 under certain conditions 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 $\mathcal{O}(nd^3 + kd \log n + kd^4)$: Line 1 uses the **insert** operation with complexity $\mathcal{O}(1)$, Line 2 calls **deletemax** with complexity $\mathcal{O}(\log n)$, and Line 3 uses **delete** and **insert**, again with complexity $\mathcal{O}(\log n)$. This complexity improves on Algorithm 2 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.

6. Experiments

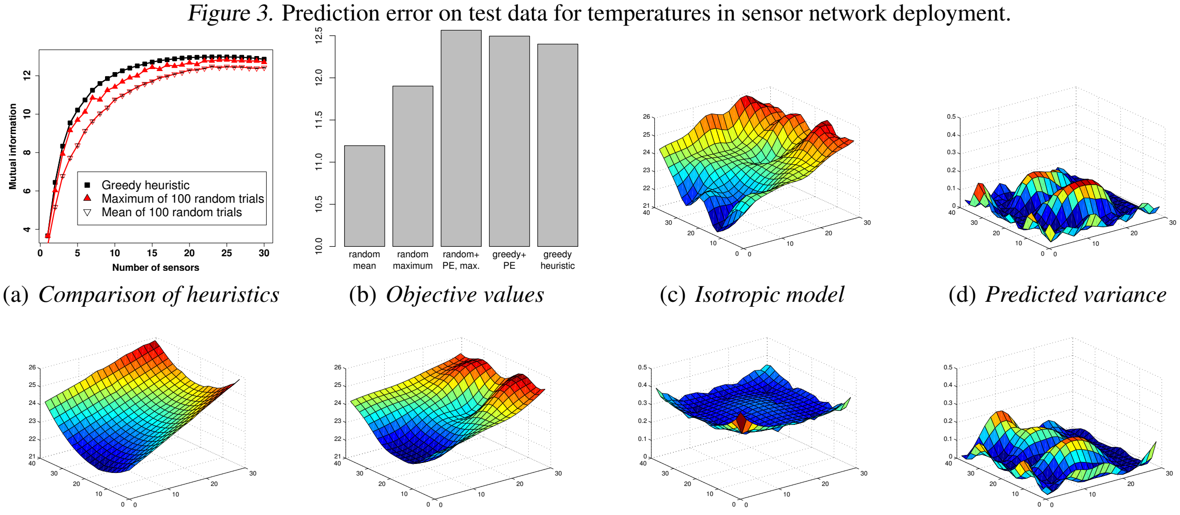
6.1. Indoor temperature measurements

In our first set of experiments, we analyzed temperature measurements from the sensor network shown in Fig. 1(a). We used our algorithms to learn informative subsets of the sensors during two times of the day: between 8 am and 9 am and between 12 pm and 1 pm. Our training data consisted of



(a) *Morning RMS* (b) *Noon RMS* (c) *Morning log-likelihood* (d) *Noon log-likelihood*

Figure 3. Prediction error on test data for temperatures in sensor network deployment.



(a) *Comparison of heuristics* (b) *Objective values* (c) *Isotropic model* (d) *Predicted variance*

(e) *Temperature (entropy)* (f) *Temperature (mutual inf.)* (g) *Variance (entropy)* (h) *Variance (MI)*

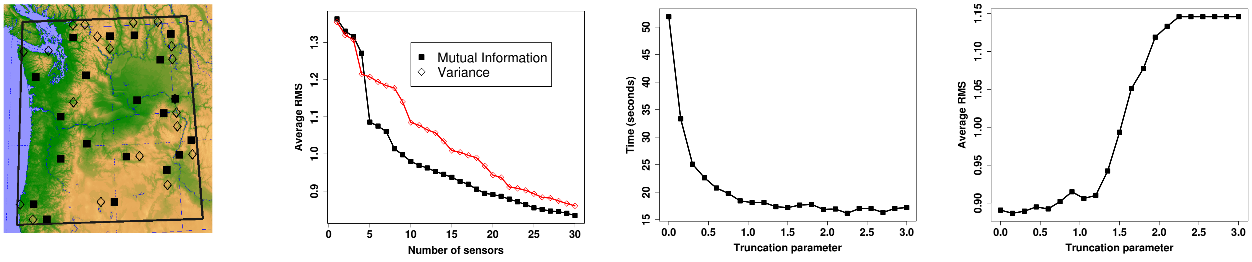
Figure 4. Comparison of predictive behavior of subsets selected using mutual information and entropy.

samples collected at 30 sec. intervals on 5 consecutive days (starting Feb. 28th 2004), the testing data consisted of the corresponding samples on the two following days.

In our initial experiments, we used an estimate of the empirical covariance matrix as the input to the algorithms. We first compared the mutual information of the sets selected by our greedy algorithm to random selections and to a hill climbing method that uses a pairwise exchange heuristic. Fig. 4(a) 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 improving the mutual information score. Fig. 4(b) 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 our algorithm). Typically, the objective values of random + PE, greedy + PE and greedy did not differ much. Note that as mentioned in Sec. 4, the performance guarantee for the greedy algorithm always provides an online approximation guarantee for the other heuristics.

Secondly, we computed the greedy subsets of sizes up to 30, using entropy and mutual information as objective functions. For testing the prediction accuracy, we provided the sensor measurements of the selected sensors from the test set, and used Gaussian inference to predict the temperature at the remaining sensor locations. We only let the algorithms choose from the locations where sensors are actually deployed in order to have test set data available for comparison. Figures 3(a) and 3(b) show the average root-mean-squares error (RMS) for these predictions, whereas Figures 3(c) and 3(d) show the average log-likelihoods. Mutual information exhibited superior performance in comparison to the entropy heuristic for increasing set sizes.

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. & Diggle, 2001). Figures 4(c) and 4(d) 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 4(e) and 4(f). Entropy tends to favor placing sensors near the boundary as observed in Sec. 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 den-



(a) *Placements of rain sensors* (b) *Precipitation RMS* (c) *Running time* (d) *Approximation error*

Figure 5. Placements (diamonds correspond to entropy, squares to MI), prediction error and running times for precipitation data.

sity. The predicted variances for each model are shown in figures 4(g) and 4(h). 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.

6.2. Precipitation data

Our second data set consisted of precipitation data collected during the years 1949 - 1994 in the states of Washington and Oregon (Widmann & Bretherton, 1999). Overall 167 regions of equal area, approximately 50 km apart, reported the daily precipitation. To ensure the data could be reasonably modelled 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 empirical covariance matrix, regularized it by adding independent measurement noise[1] of $\sigma^2 = 0.1$, and used our approximation algorithms to compute the sensor placements optimizing entropy and mutual information. We then used the test set to test prediction accuracy. Fig. 5(b) shows the average RMS prediction error. Mutual information significantly outperforms entropy as a selection criteria – often several sensors would have to be additionally placed for entropy to reach the same level of prediction accuracy as mutual information. Fig. 5(a) 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.

Fig. 5(c) shows that the computation time can be drastically decreased as we increase the truncation parameter ε from 0 to the maximum variance. Fig. 5(d) shows the RMS prediction accuracy for the 20 element subsets selected by Algorithm 2. According to the graphs, the range $\varepsilon \in [0.5, 1]$ seems to provide the appropriate trade-off between computation time and prediction accuracy.

7. Related Work

Our work falls under the rubric of discrete design. In the spatial setting (Cressie, 1991) notes that minimizing either the average or maximum predicted variance is used, but that the optimization requires evaluating $\binom{n}{k}$ placements or simulated annealing. Other methods ignore the GP, attempting to cover the space using various forms of random sampling, which are compared in (McKay et al., 1979). Pairwise exchange has been used, albeit using the Fedorov delta (Nguyen & Miller, 1992). Such algorithms often fall into local minima, and lack quality-of-solution guarantees.

Most closely related to our work is (Ko et al., 1995), which selects the best set of k sensors using the entropy (variance) criteria. Ko et. al. formulate a branch-and-bound algorithm that finds the placement corresponding to a global maxima. This approach, however, does not have a polynomial running time guarantee.

8. Conclusions

In this paper we (i) propose mutual information as a criterion for sensor placement in Gaussian processes, (ii) show the exact optimization in NP-complete, (iii) provide an approximation algorithm that is within $(1 - 1/e)$ of the maximum mutual information configuration, (iv) show how to exploit local structure in GPs for significant speed-ups. Our empirical results indicate the mutual information criteria is often better than entropy, both qualitatively and in prediction accuracy. 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.

Acknowledgements: This work was partially supported by a gift from Intel Corporation. A. Singh acknowledges the support of NSERC.

Appendix

Proof of 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 $\mathcal{U} = \{X_1, \ldots, X_n\}$. Introduce additional variables $\mathcal{S} = \{Y_1, \ldots, Y_n\}$ such that $Y_i | X_i = x \sim \mathcal{N}(x, \sigma^2)$. Note

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

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 $I(\mathcal{A}; (\mathcal{U} \cup \mathcal{S}) \setminus \mathcal{A}) = H(\mathcal{A}) - H(\mathcal{A}|(\mathcal{U} \cup \mathcal{S}) \setminus \mathcal{A}) = H(\mathcal{A}) - H(\mathcal{A}|\mathcal{U})$, because Y_i and Y_j are conditionally independent given $\mathcal{U}$. Furthermore, again because of independence, $H(\mathcal{A}|\mathcal{U})$ 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 of Lemma 4. 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 (3), 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

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

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. $H(X|\mathcal{A}) \geq H(X|\tilde{\mathcal{A}})$)

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

which concludes the argument. □

Proof of Corollary 5. The higher order terms $\mathcal{O}(\varepsilon_1^2)$ 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 $|H(X|\mathcal{A}) - H(X|\tilde{\mathcal{A}})| \leq \varepsilon$ 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 of Remark 7. We have that $H(Y|Z) < 0 \Leftrightarrow \mathcal{K}(Y, Y) + \sigma^2 - \frac{\mathcal{K}(Z,Y)^2}{\mathcal{K}(Z,Z)+\sigma^2} < \frac{1}{2\pi e}$. Using a similar argument as the proof of Lemma 4, 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 $H(Y|Z)$ and some algebraic manipulation proves the claim. □

References

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. Wiley Interscience.

Cressie, N. A. (1991). *Statistics for spatial data*. Wiley.

Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J., & Hong, W. (2004). Model-driven data acquisition in sensor networks. *VLDB*.

Golub, G., & Van Loan, C. (1989). *Matrix computations*. Johns Hopkins.

Gonzalez-Banos, H., & Latombe, J. (2001). A randomized art-gallery algorithm for sensor placement. *Proc. 17th ACM Symp. Comp. Geom.* (pp. 232–240).

Hochbaum, D. S., & Maas, W. (1985). Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, *32*, 130–136.

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

Krause, A., & Guestrin, C. (2005). *A note on the budgeted maximization of submodular functions* (Technical Report). CMU-CALD-05-103.

MacKay, D. (2003). *Information theory, inference, and learning algorithms*. Cambridge UP.

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

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

Nguyen, N.-K., & Miller, A. J. (1992). A review of some exchange algorithms for constructing discrete D-optimal designs. *Comp. Stat. and Data Analysis*, *14*, 489–498.

Paciorek, C. (2003). *Nonstationary gaussian processes for regression and spatial modelling*. Doctoral dissertation, CMU.

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

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

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

Storkey, A. J. (1999). Truncated covariance matrices and Toeplitz methods in gaussian processes. *ICANN99*.

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