NearOptimal Sensor Placements in Gaussian Processes

2008 - Andreas Krause - Ajit Singh - Carlos Guestrin
This document presents the organisation and the relevant information taken from paper.

TODO :

○ Study a Little bit more the discretization

○ Review All the citations that could be useful

I. Abstract :

When monitoring spatial phenomena, which can often be modelled 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 optimisation 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 max- imizes 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.

II. Organisation

1. Introduction
2. Gaussian Processes
    1. Modeling Sensor Data Using the Multivariate Normal

11. Conclusions

III. Notes

Notes taken on each part of the paper

1. Introduction

Monitoring spatial phenomenon + limited number of sensing devices => position of those sensors.
Modelling of sensor Measurement (what does the sensor measure in term of information in the space)

- Geometric Model

Sensing area ? Fixed sensing Radius (Gonzalez- Banos and Latombe, 2001). Not realistic : correlations between sensor measurement and actual value in the whole environment.
=> Fundamentally, the notion that a single sensor needs to predict values in a nearby region is too strong
- **Gaussian Process** Model (Cressie, 1991; Caselton and Zidek, 1984)

Weaker assumptions (more generic. == non-parametric generalization of linear regression). Learning model of the phenomenon with pilot deployment or expert knowledge
With a GP model, we asses the quality of the placement using different criterion
- Highest **Entropy** (variance). (Cressie, 1991; Shewry and Wynn, 1987)
- A-, D-, or E-optimal design
- **Mutual information**. (Caselton and Zidek (1984))

Typical sensor placement technique : greedily add sensors where uncertainty about the phenomenon is highest.
Criterion 1 : Highest Entropy: indirect criterion : measure the quality of each sensors measurement, not the prediction quality on the interesting area. Usually characterised by the sensors position that are selected to be as far from each other as possible. Border placement !
Criterion 2 : Mutual information : direct criterion : use the posterior of the GP to measure the sensor placement effect.
Optimisation : combinatorial optimization problem for maximising mutual information. NP-complete problem. mutual information is a submodular function => first approximation algorithm (in polynomial time) that guaranties a constant-factor approximation ??

2. **Gaussian Processes**

2.1. Modeling Sensor Data Using the Multivariate Normal Distribution

Set of locations V. Random variables Xv, with distribution :

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

2.2. Modeling Sensor Data Using Gaussian Processes

Interested in places where no sensor is placed => Regression techniques => GP
In a GP we consider an infinite number of observation points.
GP associated with mean function and covariance function “kernel” symmetric and semi-positive definite.

Observations xA, prediction Xy :

$$P(X_y \mid x_{\mathcal{A}}).$$

Gaussian with parameters :

$$\begin{aligned} \mu_{y|\mathcal{A}} &= \mu_y + \Sigma_{y\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}(x_{\mathcal{A}} - \mu_{\mathcal{A}}), \\ \sigma_{y|\mathcal{A}}^2 &= \mathcal{K}(y,y) - \Sigma_{y\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}\Sigma_{\mathcal{A}y}, \end{aligned}$$

2.3. Nonstationarity

How to estimate mean and kernel functions ?

- Stationary kernel : depends on difference between the locations (vector difference) (theta is a set of parameters)

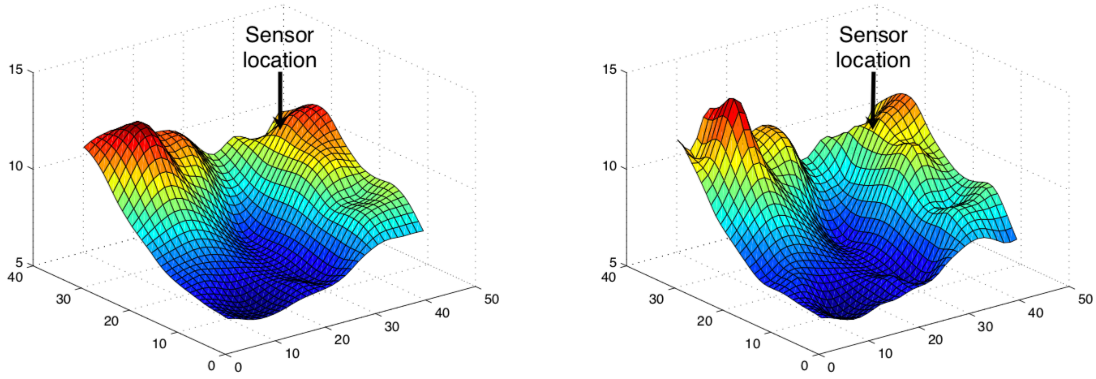
$$\mathcal{K}(u,v) = \mathcal{K}_\theta(u-v)$$

- Isotropic kernel : depends on the distance between points (Euclidian distance).

$$\mathcal{K}(u,v) = \mathcal{K}_\theta(||u -$$

Example : Exponential kernel or gaussian kernel

In the paper : kernel is neither isotropic nor stationary. Any kernel can be used, approach of Nott and Dunsmuir (2002) to compute an **empirical covariance .** The the kernel can be locally described by **collection of isotropic processes** associated with a set of reference points ? (See the paper for more information and **Section 9.2** of the paper)



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

3. Optimizing sensor placements

Objective : find the *k* best sensor locations out of a finite subset *V* of possible locations, also called **experimental design** or **sampling**

3.1 The Entropy Criterion

Intuitively, we want to place sensors which are most informative with respect to the entire design space.

Conditional entropy of the unobserved locations *V \A* after placing sensors at locations *A* :

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

minimizing this quantity aims at finding the placement which results in the lowest uncertainty about all uninstrumented locations *V \A* after observing the placed sensors *A*.

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

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

This problem is equivalent to finding a set of sensors *A* which is most uncertain about each other. It is also called D-optimal design and is a **NP hard problem**. How to solve it ? By using the **Theorem 1**

Development of a greedy heuristic (McKay et al., 1979; Cressie, 1991) :

Start from empty set of sensors A0. Greedily add placements until set's size reaches k.
Iteration : take set Ai and add the location that has the highest conditional entropy == The location with largest uncertainty given the sensors we already placed.

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

Depends **only** on the values of the covariance matrix. No difference between close-loop design (placing after measurement) versus open-loop design (placing before any measurement taken).

3.2 An Improved Design Criterion: Mutual Information

Experimentally *we observe that sensors* are placed far apart along the boundary of the space, this wastes information as there is less information to measure on the boundary. (Ramakrishnan et al. (2005)). the criterion only considers the entropy of the selected sensor locations, rather than considering prediction quality over the space of interest.

Mutual Information (Caselton and Zidek (1984))
We define :

- set of locations *V* = *S* ∪ *U*
- A set *S* of possible positions where we can place sensors,

- A set *U* of positions of interest, where no sensor placements are possible

place a set of *k* sensors that will give us good predictions at all uninstrumented locations *V* \A.

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

A∗ that maximally reduces the entropy over the rest of the space *V* \A∗.

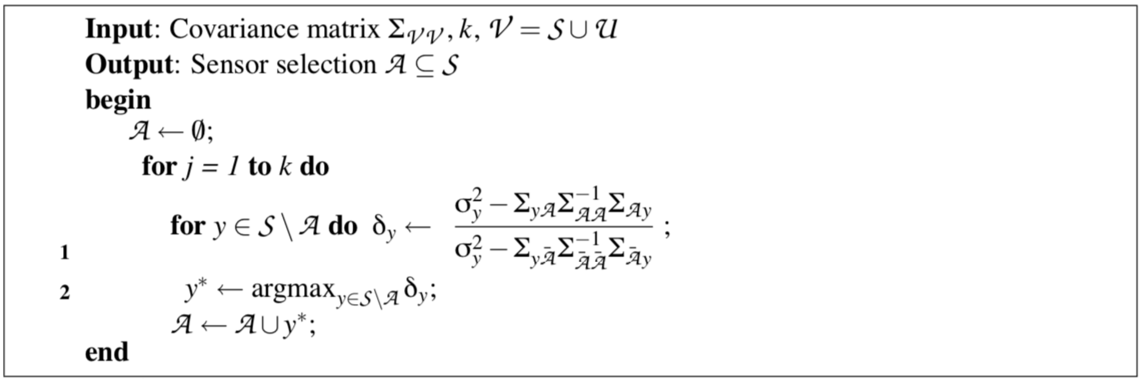
Equivalent to finding the set that maximises the *mutual information I*(*XA*;*XV\A*)

Optimizing MI is a NP hard problem. In order to approximately solve the problem in polynomial time, a greedy approximation algorithm is introduced.

4. Approximation Algorithm

4.1. The Algorithm

The algorithm is as follows :



```
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 $\delta_y \leftarrow \frac{\sigma_y^2 - \Sigma_{y\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}\Sigma_{\mathcal{A}y}}{\sigma_y^2 - \Sigma_{y\bar{\mathcal{A}}}\Sigma_{\bar{\mathcal{A}}\bar{\mathcal{A}}}^{-1}\Sigma_{\bar{\mathcal{A}}y}}$;
        $y^* \leftarrow \operatorname{argmax}_{y\in\mathcal{S}\setminus\mathcal{A}} \delta_y$;
        $\mathcal{A} \leftarrow \mathcal{A} \cup y^*$;
end
```

Algorithm 1: Approximation algorithm for maximizing mutual information.

At each step we maximise the following quantity (we want to add the sensor that adds a maximum information)

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

$$= H(y \mid \mathcal{A}) - H(y \mid \bar{\mathcal{A}}),$$

We see that the greedy rules for the Entropy criterion considered only

the uncertainty of the sensor with respect to the placements A (sensors already in place). Here the MI criterion takes into account the placement of the sensor y with respect to the Â points. If forces to pick a y that is "central" wrt to Â (conditional entropy is minimal).
Recap : H(y|A) : maximise the uncertainty of the placement wrt A : sensors placed
H(y|Â) : minimise the uncertainty of the placement wrt Â : positions where there is no sensor

4.2. An Approximation Bound

If the grid is fine enough, the bound of optimal sensors positions found by the greedy algorithm is of 63% (~ 1 - 1/e).
For a returned set Â. The mutual information is at least (for some small ε > 0)

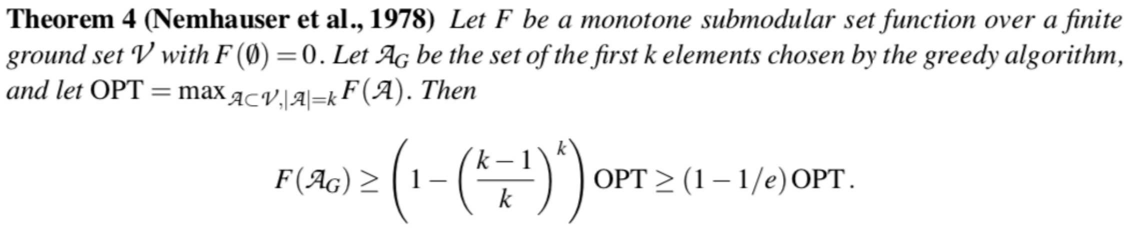
$$\mathrm{MI}(\widehat{A}) \geq (1-1/e) \max_{\mathcal{A}\subset\mathcal{S},|\mathcal{A}|=k} \mathrm{MI}(\mathcal{A}) - k\varepsilon,$$

To prove that the **submodularity** is used : a function *F* is called *submodular*, if for all *A*,*B* ⊆ *V* it holds that *F*(*A* ∪ *B*) + *F* (*A* ∩ *B*) ≤ *F* (*A*) + *F* (*B*).
An other formulation : *A* ⊆ *A* ′ ⊆ *V* and *y* ∈ *V* \ *A* ′ it holds that *F* (*A* ∪ *y*) − *F*(*A*) ≥ *F*(*A*′ ∪ *y*) − *F*(*A*′).
Intuition : diminishing returns. Adding a sensor to a large set is less rewarding that adding a sensor to a small set.

By using the fact that "the information never hurts". *H*(*y* | *A*) ≥ *H*(*y* | *A* ∪*B*) (Cover and Thomas, 1991), We have that :

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

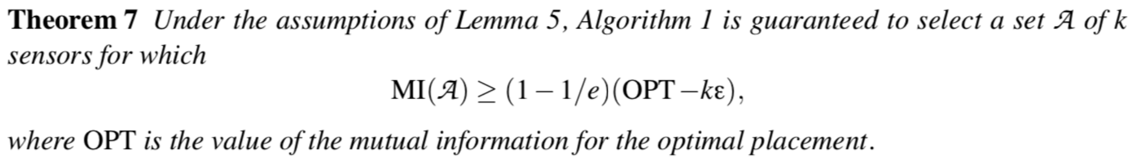
In this context a monotonic function is such as : *F* (*A* ∪ *y*) ≥ *F* (*A*)



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*

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

After some developments we have the final theorem :



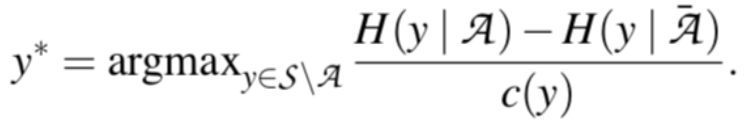
Theorem 7 *Under the assumptions of Lemma 5, Algorithm 1 is guaranteed to select a set $\mathcal{A}$ of k sensors for which*

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

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

4.3. Sensor Placement with Non-constant Cost Functions

The cost of placing sensors depend on the specific location. We consider a budgeted where each sensor chosen has a cost.
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:

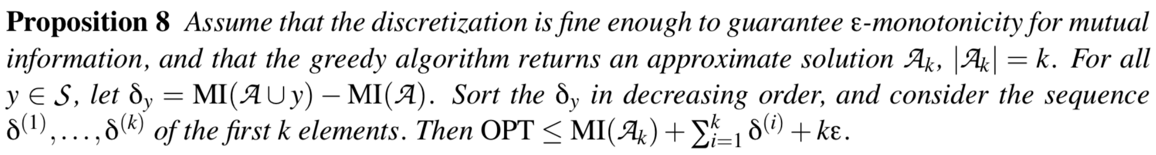


$$y^* = \mathrm{argmax}_{y \in \mathcal{S} \setminus \mathcal{A}} \frac{H(y \mid \mathcal{A}) - H(y \mid \bar{\mathcal{A}})}{c(y)}.$$

Krause and Guestrin (2005) show that this algorithm achieves an approximation guarantee of
(1 – 1/*e*) OPT – 2*L*ε , *c*min
where *L* is the available budget, and cmin is the minimum cost of all locations.
We can also define some discretisation levels similarly as in Corollary 6.

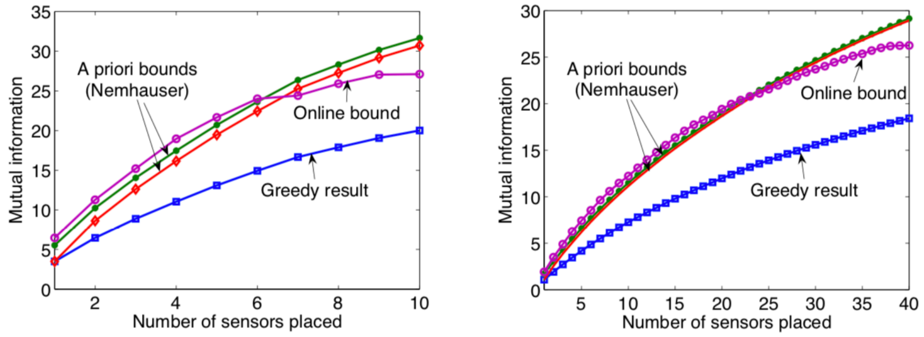
4.4. Online Bounds

For most practical problems however, this bound is very loose. So we are able to compute **online bounds** to the optimal value (“live” bounds).



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* $\mathrm{OPT} \leq \mathrm{MI}(\mathcal{A}_k) + \sum_{i=1}^{k} \delta^{(i)} + k\varepsilon$.

especially for large placements, this bound can be much tighter than the bound guaranteed by Theorem 7



(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.

4.5. Exact Optimization and Tighter Bounds Using Mixed Integer Programming

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) . See the paper for more results.

5. Scaling Up

Motivation :

Greedy update :

- Entropy maximisation : *H*(*y* | *A*) complexity of this operation is *O*(*k*3)
- Mutual Information maximisation : *H*(*y* | *A*) requiring *O*(*n*), for *n* = |*V* |.

We need to recompute the score of all possible locations at every iteration, so the complexity for the selection *k* sensors is *O*(*kn*4)

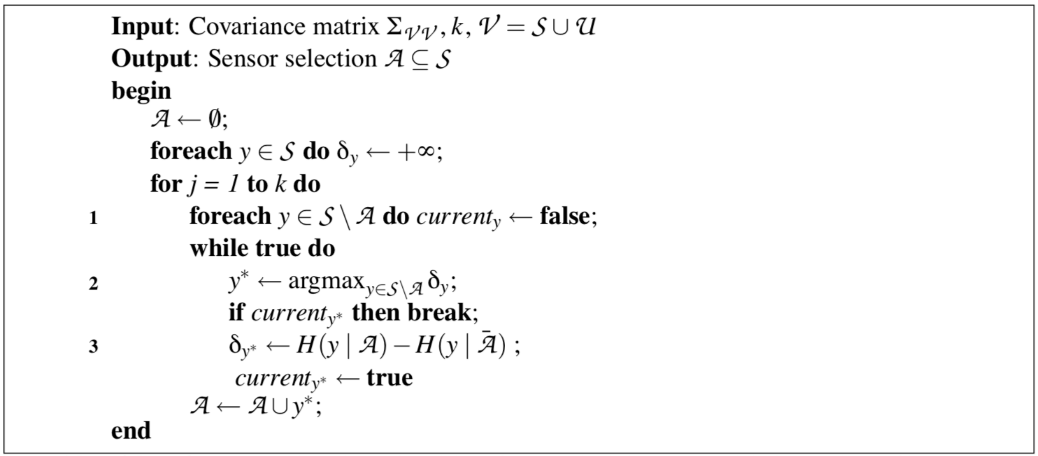
=> Lazy Evaluation : *O*(*kn*3)

=> Local Kernels : O(kn)

5.1. Lazy Evaluation Using Priority Queues

The idea is to evaluate for each step as few as possible positions.

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



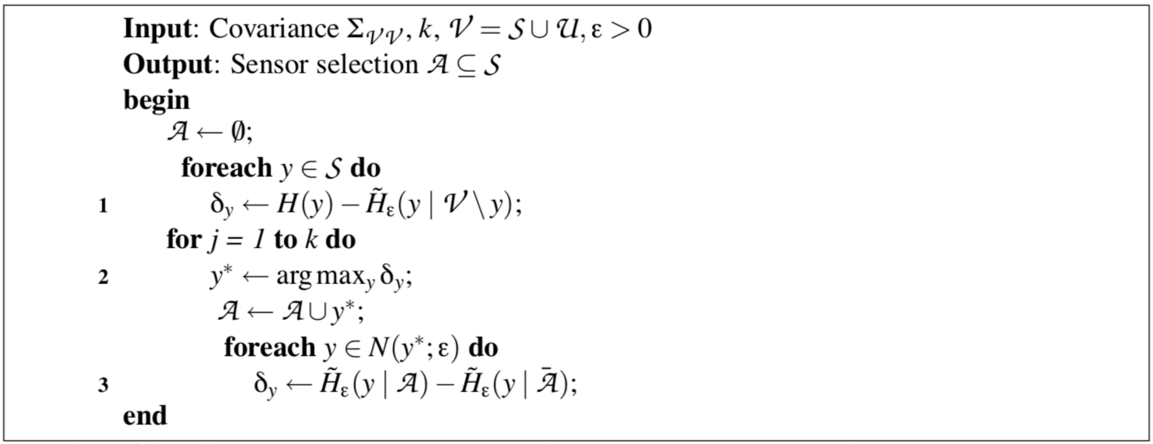
```
Input: Covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, $k$, $\mathcal{V} = \mathcal{S} \cup \mathcal{U}$
Output: Sensor selection $\mathcal{A} \subseteq \mathcal{S}$
begin
    $\mathcal{A} \leftarrow \emptyset$;
    foreach $y \in \mathcal{S}$ do $\delta_y \leftarrow +\infty$;
    for $j$ =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

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

5.2. Local Kernels

The idea is to use the fact that in GP variables which are far apart are actually independent.
It can be modelled with Kernel function which as compact support (non-zero for small portion the space).
OR it can be interesting to remove the points where : $|K(x,y)| \leq \varepsilon$.
This can be justified as in the paper by considerations on the bounds of the decrease in entropy when removing such points.

This truncation allows to compute $H(y \mid \hat{A})$ much more efficiently, at the expense of this small absolute error.



```
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
        $\delta_y \leftarrow H(y) - \tilde{H}_\varepsilon(y \mid \mathcal{V} \setminus y)$;
    for $j$ = 1 to $k$ do
        $y^* \leftarrow \arg\max_y \delta_y$;
        $\mathcal{A} \leftarrow \mathcal{A} \cup y^*$;
        foreach $y \in N(y^*; \varepsilon)$ do
            $\delta_y \leftarrow \tilde{H}_\varepsilon(y \mid \mathcal{A}) - \tilde{H}_\varepsilon(y \mid \bar{\mathcal{A}})$;
end
```

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

Here, $\tilde{H}\varepsilon$ refers to the truncated computation of entropy
$N(y_*;\varepsilon) \leq d$ refers to the set of elements $x \in S$ for which $|K(y_*,x)| > \varepsilon$.

This decreases drastically the complexity. (See paper)
In order to further inprove the complexity, we can use a Relaxed Heaps Data Structure.
Can also be combined with lazy evaluation previously described.

6. Robust Sensor Placements

6.1. Robustness Against Failures of Nodes
6.2. Robustness Against Uncertainty in the Model Parameters

7. Related Work

1. Objective Functions
2. Optimization Techniques
3. Related Work on Extensions
4. Related Work in Machine Learning
5. Relationship to Previous Work of the Authors

8. Notes on Optimizing Other Objective Functions

1. A Note on the Relationship with the Disk Model
2. A Note on Maximizing the Entropy
3. A Note on Maximizing the Information Gain
4. A Note on Using Experimental Design for Sensor Placement

9. Experiments

1. Data Sets
2. Comparison of Stationary and Non-stationary Models
3. Comparison of Data-driven Placements with Geometric Design Criteria
4. Comparison of the Mutual Information and Entropy Criteria
5. Comparison of Mutual Information with Classical Experimental Design Criteria
6. Empirical Analysis of the Greedy Algorithm
7. Results on Local Kernels