Grey-Box Bayesian Optimization for Sensor Placement in Assisted Living Environments

Shadan Golestan, Omid Ardakanian, Pierre Boulanger

Computing Science Department, University of Alberta
{golestan, ardakanian, pierreb}@ualberta.ca

Abstract

Optimizing the configuration and placement of sensors is crucial for reliable fall detection, indoor localization, and activity recognition in assisted living spaces. We propose a novel, sample-efficient approach to find a high-quality sensor placement in an arbitrary indoor space based on grey-box Bayesian optimization and simulation-based evaluation. Our key technical contribution lies in capturing domain-specific knowledge about the spatial distribution of activities and incorporating it into the iterative selection of query points in Bayesian optimization. Considering two simulated indoor environments and a real-world dataset containing human activities and sensor triggers, we show that our proposed method performs better compared to state-of-the-art black-box optimization techniques in identifying high-quality sensor placements, leading to accurate activity recognition in terms of F1-score, while also requiring a significantly lower number (51.3% on average) of expensive function queries.

Introduction

Smart indoor spaces are commonly equipped with various networked sensors, such as motion sensors and cameras (Cook et al. 2012), to monitor activities and physical and mental health of the occupants, enabling caregivers to detect and respond quickly to critical events, such as falls. This is crucial as 1 in 4 older adults report falling per year (Centers for Disease Control and Prevention 2020).

The placement of sensors in the environment is one of the main factors determining the performance of machine learning models that utilize the data they generate (Yang 2021). In particular, an optimized sensor placement strategy makes possible accurate activity recognition and localization using the smallest number of sensors. However, due to the exponential size of the search space and the high cost of evaluating the model performance for each sensor placement under normal occupancy conditions, finding the best sensor placement through exhaustive search is prohibitive in practice.

To date, many efforts have been made to design sample-efficient techniques to optimize the location of sensors, given a lower bound on the performance of downstream applications that consume the sensor data. Greedy and evolutionary algorithms, such as the genetic algorithm (GA), are the most notable methods used to place and configure sensors in an indoor environment (Thomas, Crandall, and Cook 2016; Wu et al. 2020; Yu et al. 2020). But these methods do not perform well in general, because they rely only on local information provided by the samples of f, the function being optimized (Mori, Takeda, and Matsumoto 2005), with x being a sensor placement and $f(x)$ representing some performance measure of the machine learning model. On the contrary, estimation of distribution algorithms, such as Bayesian Optimization (BO) (Shahriari et al. 2015), use local information to acquire global information about $f(x)$, i.e., building a probabilistic surrogate model of this function. When this model represents $f(x)$ accurately, the optimizer can perform more effective exploration/exploitation. A wide range of problems have been recently solved using BO, from optimization over permutation spaces (Deshwal et al. 2022) and combinatorial spaces (Deshwal et al. 2020, 2023) to setting up sensor networks for air quality monitoring (Hellan, Lucas, and Goddard 2022). However, the application of BO to optimize the sensor placement for indoor activity recognition has not been explored in previous work.

The main shortcoming of BO is that it treats $f(x)$ as a black-box function, meaning that it analyzes only its input-output behavior, disregarding any inherent, domain-specific knowledge that might exist about this function. Grey-box optimization (Astudillo and Frazier 2021), however, incorporates this knowledge in the optimization process. Although it requires more careful design, it can be faster and drastically improve the quality of the solution. Our hypothesis is that in the problem of optimizing sensor placement with respect to the performance of an activity recognition model, inherent knowledge about the distribution of activities in different parts of the space could help BO quickly identify important regions in the search space.

This paper presents Distribution-Guided Bayesian Optimization (DGBO), a novel grey-box BO algorithm that learns the spatial distribution of activities and takes advantage of this knowledge to speed up the search for the best sensor placement. This algorithm, depicted in Fig. 1, combined with high-fidelity building simulation for generating realistic movement and activity traces and corresponding sensor triggers, establishes the foundation of a framework for sample-efficient identification of a motion sensor placement that supports accurate activity recognition in an aged-

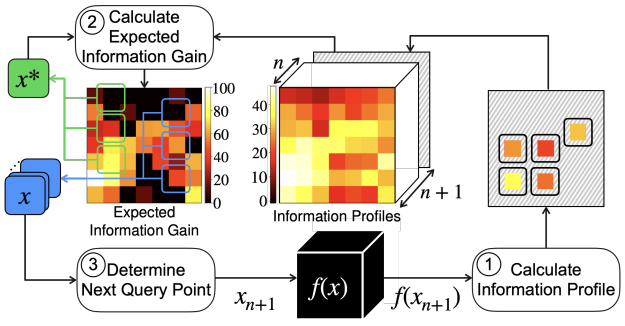


Figure 1: An overview of our proposed grey-box optimization algorithm (DGBO).

care facility. The proposed approach does not cause any discomfort for the occupants, nor does it infringe their privacy or raise major ethical concerns. We show empirically that DGBO outperforms BO, genetic, and greedy algorithms in two simulated test buildings, and corroborate this finding using real sensor data collected in an indoor environment with multiple installed sensors. Our evaluation result indicates that the surrogate model of the objective function contains useful information in the context of assisted living. By leveraging this information and the inherent knowledge of the indoor space, DGBO achieves superb performance with respect to the activity recognition accuracy and better sample efficiency than BO across all test environments.

Our contribution is threefold: 1) We propose DGBO: a novel grey-box BO algorithm for optimizing sensor placement in an indoor environment; 2) We develop a simulation-based assessment and optimization framework allowing the use of synthetic activities and movement trajectories to evaluate the black-box function, instead of real-world traces that are difficult and expensive to collect; 3) Through extensive experiments, we show the efficacy of DGBO in finding high-quality sensor placements in three different indoor environments. Our dataset and code are available on GitHub.[1]

Related Work

Maximizing area coverage of sensors. Most related work on sensor placement seeks to maximize the area covered by the sensors, without considering the accuracy of downstream applications that consume the sensor data (see for example (Fanti et al. 2017; Gungor, Rosing, and Aksanli 2020)). The fundamental limitation of this approach comes from the assumption that all areas of the building are equally important and should be monitored for an application such as activity recognition. In the real world, some building spaces are rarely occupied, which implies that covering the *active* regions is more important. To address this shortcoming, several studies utilize contextual information about the building (e.g. its floor plan) to obtain a set of points that represent the potential location of occupants, and place sensors such that these points are maximally covered (Barry and Thron 2019; Vlasenko, Nikolaidis, and Stroulia 2014; Wu et al. 2020). For instance, Vlasenko *et al.* (2014) designed a near-optimal greedy algorithm that gets synthetic occupant trajectories and finds a motion sensor placement for accurate indoor localization. However, assuming all points are equally important may lead to misidentification of critical, yet rare activities, e.g., bathing, in an aged-care monitoring application. We use a greedy algorithm as a baseline.

Maximizing the activity recognition accuracy. There are only a few known studies that identify the best sensor placement by maximizing the accuracy of a downstream application. Thomas *et al.* (2016) proposed a framework to find the optimal placement of omnidirectional, ceiling-mounted motion sensors by maximizing the accuracy of an activity classifier while minimizing the number of deployed sensors, given the occupants' activities and movement trajectories. They used a real-world dataset to generate synthetic occupant trajectories and employed GA to effectively find high-quality sensor placements. However, generating a synthetic dataset from real-world sensor readings is costly and challenging. GA serves as the baseline for comparison.

Gaussian Processes (GP)-based sensor placement. Optimal sensor placements for monitoring spatial phenomena that can be modeled as GP, e.g., temperature, has received some attention (Andersson et al. 2023). Guestrin et al. (2005) proposed a near-optimal algorithm for finding a placement that maximizes the mutual information between the placement and the rest of the space. While this work shares similarities with ours, incorporating domain knowledge into this framework remains an open problem.

Sample-efficient optimization of black-box functions. In recent work, Deshwal *et al.* (2023) proposed a surrogate modeling approach for high-dimensional combinatorial spaces. They used a dictionary-based embedding to map discrete structures from the input space into an ordinal feature space, allowing the use of continuous surrogate models, such as the Gaussian process. However, the size of the search space grows exponentially with the number of possible sensor locations (ranging from nearly 200 to 1000 locations in a small, e.g., 700-square-foot, indoor environment). Due to its inability to directly control the sensor numbers, our assessments show that applying this method to the sensor placement problem often leads to installing >500 sensors in our environments, which is overly costly, rendering this approach of limited practical value in this domain.

Grey-box Bayesian optimization. Some studies have explored using the domain-specific knowledge available about $f(x)$, thereby treating $f(x)$ as a grey-box function (Astudillo and Frazier 2021). They have sought a method to capture this knowledge and incorporate it into the definition of an *acquisition function*, which decides on the next query point in the optimization process. For example, Weissteiner *et al.* (2023) proposed a method for estimating an upper uncertainty bound, to design a new acquisition function for BO-based combinatorial auctions optimization.

Novelty of this work. We cast sensor placement in an indoor environment as a Bayesian optimization problem. We then introduce a novel grey-box BO, called DGBO, and compare the resulting motion sensor placement with those found by BO, GA, and greedy. We show that the available domain-specific knowledge in our problem guides DGBO towards better sensor placements, and increases its sample efficiency.

[1] https://github.com/shadangolestan/SensorConfigOptimization

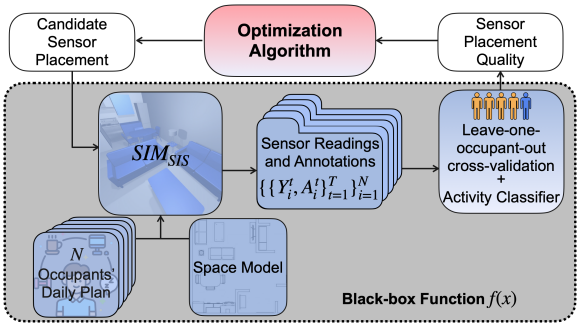


Figure 2: The proposed framework for sensor placement.

Methodology

We present our simulation-driven motion sensor placement framework in Fig. 2. At the core of this framework lies an optimization algorithm that finds a sensor placement maximizing the performance of a model that classifies *activities of daily living* (ADL), given data generated by the sensors. To efficiently obtain motion sensor data for each candidate sensor placement, we use a simulator of smart indoor spaces (SIM_{sis}) developed by Golestan *et al.* (2020) (such simulators are thoroughly studied (Briscoe et al. 2022; Dahmen and Cook 2019)). The SIM_{sis} gets ADL plans of N occupants and the location of motion sensors, and simulates motion sensor triggers. The sensor triggers and corresponding activities are then passed to the activity classifier module to obtain the result. This module trains an ADL classifier using 80% of the dataset, and evaluates its performance on the remaining 20% of the dataset (test set). Putting all together, we treat the simulation of the indoor environment as a stochastic function, $f(x)$, that gets as input a candidate sensor placement x and outputs a noisy observation, which is a measure of performance of an activity recognition model.

Simulator of Smart Indoor Spaces

The SIM_{sis} is an open-source (Golestan 2020) indoor environment simulator capable of realistically modeling occupants' behavior and simulating motion sensor readings. It receives a space model, i.e., the specification of an indoor environment, daily plans of N occupants i.e., regular ADLs, such as sleeping and cooking, and a candidate sensor placement which describes the number and locations of motion sensors. Each sample of the model generates a slightly different daily plan in terms of ADL order and duration. The SIM_{sis} generates a synthetic dataset with sensor readings and their corresponding activity for each of the N occupants:

$$\mathcal{D} = \{\{Y_i^t, A_i^t\}_{t=1}^T\}_{i=1}^N \tag{1}$$

where Y_i^t and A_i^t denote respectively the sensor triggers (a binary vector of size D where D is the number of sensors installed in the environment) and the activity of the i-th occupant at time t. An ideal sensor placement[2] should result in a dataset where each activity has a distinct enough pattern in the space of sensor readings so that an activity recognition model can recognize it (Freedman and Zilberstein 2019).

[2]We assume the motion sensors are omnidirectional and ceiling-mounted, reducing the placement to a 2D positioning task.

Activity Classifier

The activity classifier gets the dataset $\mathcal{D}$ generated by a fixed set of sensors installed in the environment and applies the leave-one-out cross-validation method. Specifically, it considers the data that pertains to one occupant as the test set and the data of the remaining occupants as the training set. This process is repeated for each occupant to obtain different test sets. Following Cook *et al.* (2012), we use a random forest as the activity recognition model. For each test set, the activity classifier calculates the macro-averaged F1-score of the ADL and then outputs the arithmetic mean of the F1-scores (given below) as the model performance:

$$F^1 = \frac{1}{N}\Sigma_{i=1}^N \frac{1}{M}\Sigma_{j=1}^M \frac{tp_j}{tp_j + \frac{1}{2}(fp_j + fn_j)} \tag{2}$$

Here N is the number of occupants, M is the number of activities, and tp_j, fp_j, and fn_j are true positive, false positive and false negative of the j-th activity, respectively. Notice that F^1 is sensitive to both false negative and false positive. We argue that it is a good performance measure, because both of these factors are important in the activity recognition task, and in particular for elderly monitoring systems where failing to alert caregivers could have dire consequences and false alarms strain caregivers. Eqn. 2 calculates the macro-averaged F1-score over the M activities, since there are rare and short, but precarious activities, e.g., bathing, that should be deemed as important as other activities in our problem.

Optimization Algorithm

The optimization algorithm module receives an observation, which is the performance of the activity recognition model expressed in Eqn. 2 ($f(x)$=F^1), and proposes a candidate sensor placement in each iteration to maximize this performance measure. This optimization can be performed using vanilla BO, DGBO (our proposed method), greedy and genetic algorithms, which are two popular heuristic search algorithms adopted in the literature.

Problem Formulation Let us denote the possible sensor locations in the indoor environment as a set $\mathcal{L}$ define as:

$$\mathcal{L} = \{l_i\}_{i=1}^L \tag{3}$$

We seek to find the subset of $\mathcal{L}$ such that if motion sensors are installed at these locations then $f(x)$ will be maximized.

Bayesian Optimization BO is a sequential search strategy for optimizing a black-box function. In our problem, this function is f which returns the performance of the activity recognition model. Thus, BO solves the following problem:

$$\max_{x \in S} f(x) \tag{4}$$

where S is the search space containing all possible placements of D sensors ($1 \leq D \leq L$) and x represents a feasible sensor placement. Note that the size of S grows exponentially with the number of sensors: $|S| = \binom{L}{D}$. The BO assumes that observations of $f(x)$ are the results of a stochastic process and are independent. There are two sources of noise in f: 1) SIM_{sis}: it does not necessarily capture all dynamics and uncertainties that exist in the real-world; 2) the

activity classifier: the random forest fits several decision tree classifiers on different randomly generated sub-samples of the dataset. Thus, each execution might yield slightly different results. BO approximates f using a surrogate model, denoted $\hat{f}$, given a set of observations $\mathcal{Z}$:

$$\hat{f}(x) = p(f(x)|\mathcal{Z}). \tag{5}$$

According to this definition, the surrogate model $\hat{f}$ estimates F[1] (Eqn. 2) given a sensor placement x. We use the Probabilistic Random Forest (PRF) as the BO's surrogate model (Hutter, Hoos, and Leyton-Brown 2011). We use the observations to compute the mean and standard deviation of $\hat{f}(x)$ at a sensor placement x and denote them as μ_x and σ_x. Hence, we can write $\hat{f}(x)=\mu_x+\sigma_x u$ where $u\sim\mathcal{N}(0,1)$. In each iteration n, the surrogate model is indeed a prior over f given observations received from the first iteration to the n-th iteration: $\mathcal{Z}_{1:n}=\{(x_i, f(x_i)\}_{i=1}^{n}$.

The surrogate model $\hat{f}$ is used to form an acquisition function $\alpha(x;\hat{f})$, which determines the next candidate sensor placement (x_{n+1}) to evaluate in iteration $n+1$. We use the expected improvement (EI) function[3] (Frazier 2018):

$$\alpha_{\text{EI}}(x;\hat{f}) = \int_{-\infty}^{\infty} \max\big(\hat{f}(x) - f(x^*), 0\big)\phi(u)\,du = \\ (\mu_x - f(x^*))\Phi(\frac{\mu_x - f(x^*)}{\sigma_x}) + \sigma_x\phi(\frac{\mu_x - f(x^*)}{\sigma_x}) \tag{6}$$

where $f(x^*)=\max\{f(x_1), ..., f(x_n)\}$ is the observation that corresponds to the best sensor placement found so far x^* (i.e. the incumbent); $\Phi(.)$ and $\phi(.)$ denote the cumulative density function (CDF) and probability density function (PDF) of the standard normal distribution, respectively. The acquisition function is used to find potentially better sensor placements. The next query point is given by:

$$x_{n+1} = \arg\max_x \alpha_{\text{EI}}(x;\hat{f}), \tag{7}$$

which is the point with the largest expected improvement. The choice of EI is due to its widespread use and because it strikes a good balance between exploration and exploitation.

After evaluating x_{n+1}, the corresponding observation, i.e., $z_{n+1}=(x_{n+1}, f(x_{n+1}))$, is appended to the sequence of past observations, $\mathcal{Z}_{1:n}$, to obtain $Z_{1:n+1}=\{Z_{1:n}, z_{n+1}\}$. Next, $\mathcal{Z}_{1:n+1}$ is used to update the surrogate model $\hat{f}$, resulting in the posterior surrogate model that better approximates f. The posterior surrogate model is then used as a prior for obtaining x_{n+2}. The process continues for 1000 iterations and the best sensor placement found is reported. We randomly choose the initial placement (x_1) and implement BO using the package from (Li et al. 2021).

Distribution-Guided Bayesian Optimization (DGBO) We now describe the proposed grey-box BO. Similar to BO, DGBO utilizes a surrogate model and an acquisition function. It also takes advantage of the distribution of indoor activities, which can be captured from the evaluation of $f(x)$.

[3]Our initial experiments show that EI performs better than other alternatives, e.g., Probability of Improvement and Upper Confidence Bound.

For each possible sensor location $l_i\in\mathcal{L}$, we define an activation region R_i which contains every point that would be within the range of a motion sensor installed at l_i. Note that the activation regions of two sensors may overlap. The objective is to estimate an unknown function $I(R_i)$ that outputs the amount of information R_i provides. This function provides insight into how useful a sensor is when monitoring a region. Intuitively, this should depend on the spatial distribution of activities and the location of other sensors in the environment. To obtain prior information about each activation region R_i, we query $f(x)$ at l_i; this is equivalent to installing exactly one sensor in the middle of R_i and using only its data for activity recognition. DGBO uses the prior information about each activation region R_i to measure the amount of information that this region could provide at iteration n, and inserts it into a tabular data structure, denoted $\mathcal{I}_{0:n}\{R_i\}$ and marked *Information Profiles* in Fig. 1. Thus, after receiving the n-th observation ($z_n=(x_n, f(x_n))$), we update $\mathcal{I}_{0:n}\{R_i\}$ for those R_is that have a sensor in x_n (see module 1 in Fig. 1) :

$$\mathcal{I}_{0:n}\{R_i\} = \left\{\mathcal{I}_{0:n-1}\{R_i\}, \frac{\mathcal{I}_{0:0}\{R_i\}}{\Sigma_{l_j\in x_n}\mathcal{I}_{0:0}\{R_j\}} f(x_n)\right\} \tag{8}$$

where $\mathcal{I}_{0:0}\{R_i\}$ is the prior information of region R_i. Eqn. 8 contributes a portion of $f(x_n)$ to R_i based on its prior information relative to other regions' prior information. This is similar to the temporal credit assignment problem (Sutton 1984) in Reinforcement Learning, wherein a sequence of actions taken by an agent receive credit based on its outcome. With this analogy, our approach, which is a form of spatial credit assignment, identifies the contribution of a number of regions to the particular level of performance achieved as a result of installing sensors in the middle of these regions. We use Eqn. 8 to build a prior over $I(R_i)$:

$$\hat{I}(R_i) = p(I(R_i) \mid \mathcal{I}_{0:n}\{R_i\}). \tag{9}$$

Assuming $\hat{I}(R_i)$ follows a Gaussian distribution ($\mathcal{N}(\mu_{R_i}, \sigma_{R_i})$), we can use the mean and standard deviation of $\mathcal{I}_{0:n}\{R_i\}$ to calculate its parameters. We define the relative *information gain* of a region compared to the information gain of the incumbent:

$$I_n^+(R_i) = \max(\hat{I}(R_i) - I^*, 0) \tag{10}$$

where $I^*=\frac{1}{D}\sum_{l_j\in x^*} I_{n-1}^+(R_j)$ is the average information gain of regions in incumbent sensor placement x^*, and $I_0^+(R_i)=\mathcal{I}_{0:0}\{R_i\}$. The expected information gain of each region is then calculated by taking the expected value of Eqn. 10 (see module 2 in Fig. 1).

For each placement x, our acquisition function, $\alpha_{\text{DG}}(x)$, calculates the average expected information gain of the regions R_i where a sensor is placed in x:

$$\alpha_{\text{DG}}(x;\hat{f}) = \frac{1}{D}\sum_{l_i\in x} \mathbb{E}[I_n^+(R_i)] \tag{11}$$

With a reparameterization, we can write $\hat{I}(R_i)=\mu_{R_i}+\sigma_{R_i}u$ where $u\sim\mathcal{N}(0,1)$:

$$\alpha_{\text{DG}}(x;\hat{f})=\frac{1}{D}\sum_{l_i\in x}\int_{u_0}^{+\infty}(\mu_{R_i}+\sigma_{R_i}u-I^*)\phi(u)\,du$$

$$=\frac{1}{D}\sum_{l_i\in x}\left(\int_{u_0}^{+\infty}(\mu_{R_i}-I^*)\phi(u)\,du+\int_{u_0}^{+\infty}\sigma_{R_i}u\phi(u)\,du\right)$$

$$=\frac{1}{D}\sum_{l_i\in x}\left((\mu_{R_i}-I^*)\int_{u_0}^{+\infty}\phi(u)\,du+\frac{\sigma_{R_i}}{\sqrt{2\pi}}\int_{u_0}^{+\infty}ue^{\frac{-u^2}{2}}\,du\right)$$

$$=\frac{1}{D}\sum_{l_i\in x}\left((\mu_{R_i}-I^*)(1-\Phi(u_0))-\frac{\sigma_{R_i}}{\sqrt{2\pi}}\left[e^{\frac{-u^2}{2}}\right]_{u_0}^{+\infty}\right)$$

$$=\frac{1}{D}\sum_{l_i\in x}(\mu_{R_i}-I^*)\Phi(\frac{\mu_{R_i}-I^*}{\sigma_{R_i}})+\sigma_{R_i}^2\phi(\frac{\mu_{R_i}-I^*}{\sigma_{R_i}}) \quad (12)$$

where $u_0=\frac{I^*-\mu_{R_i}}{\sigma_{R_i}}$ To exploit the exploitation and exploration tendency of α_{DG} and α_{EI}, respectively, we use scalarization and maximize $\alpha_{\text{DG}}(x;\hat{f})+\alpha_{\text{EI}}(x;\hat{f})$ to choose the next query point (see module 3 in Fig. 1). Intuitively, this point maximizes the sum of expected improvement and expected information gain, rather than any of them.

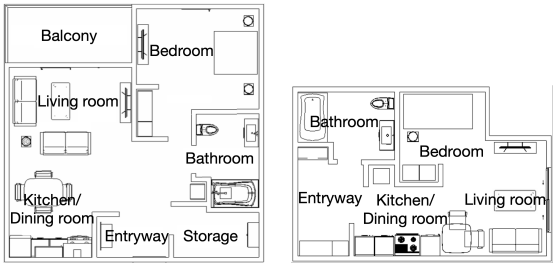
Experiments

Simulation Case Study

We consider two testbeds for our simulation case study: an 8×8 (m^2) fully-furnished, one-bedroom suite (marked T1) and a 5.2×8 (m^2) fully-furnished, studio suite (marked T2) from the Lifestyle Options Retirement Communities (Lifestyle Options 2023). Fig. 3 shows the floor plan of these suites. The suites are designated for older adults needing independent living, assisted living, or memory care. We choose these suites as our testbeds due to their differences, such as layout and size, which could lead to a diverse set of traces and unique challenges. Specifically, T2 exhibits distinct movement trajectories in some areas, e.g., the kitchen, since the entryway and bathroom are accessible only from the kitchen, and T1 exhibits slightly scattered movement trajectories and has no trajectories in the balcony.

Similar to most related work, such as (Thomas, Crandall, and Cook 2016), we postulate that the set of possible sensor locations forms a 2D grid on the given floor plan. Thus, $L=H\times W$ (see Eqn. 3) with $H=\lceil h/\epsilon\rceil-1$ and $W=\lceil w/\epsilon\rceil-1$ where h and w are the height and width of the indoor environment, respectively, and ϵ is the spacing between consecutive rows and columns. Higher ϵ values indicate lower granularity and lower computation overhead.

The case study includes five occupants ($N=5$) performing various ADLs independently in each testbed. These activities and corresponding sensor triggers are simulated by SIM_{sis}. Table 1 shows an ordered list of 23 detailed activities the occupants perform in 196 minutes in total (in Eqn. 1, A is the list of detailed activities and $T=\frac{196\times60}{3}$ since data is collected every 3 seconds). The order of performing ADLs and some detailed activities in each ADL (denoted using the same superscript in Table 1) can be interchanged, producing slightly different activity plans.



(a) Testbed 1 (T1). (b) Testbed 2 (T2).

Figure 3: The floor plan/space model of the two apartments: (a) an $8\times8(m^2)$ one-bedroom suite; (b) a $5.2\times8(m^2)$ studio.

ADL	Seq. of detailed activities (duration in minutes)
Bathing	Undress (5), Take a shower (15), Dress (6)
Hygiene	Use toilet (3), Wash hands (3)
Dining routine	Make tea (10), Grab ingredients (2), Fry eggs (10), Toast breads (5), Grab utensils (1), Eat (10), Take medicine (2), Wipe dining table^a (5), Wash dishes^a (3), Clean kitchen^a (5)
Brooming	Grab the broom from storage (2), Broom (7), Return the broom (2)
Others	Sit and work with tablet^b (30), Exercise^b (30), Watch TV^b (15), Iron^b (5), Sleep^b (20)

Table 1: ADLs in our simulated case study. ADLs and detailed activities with the same superscript can be shuffled.

Real-World Case Study

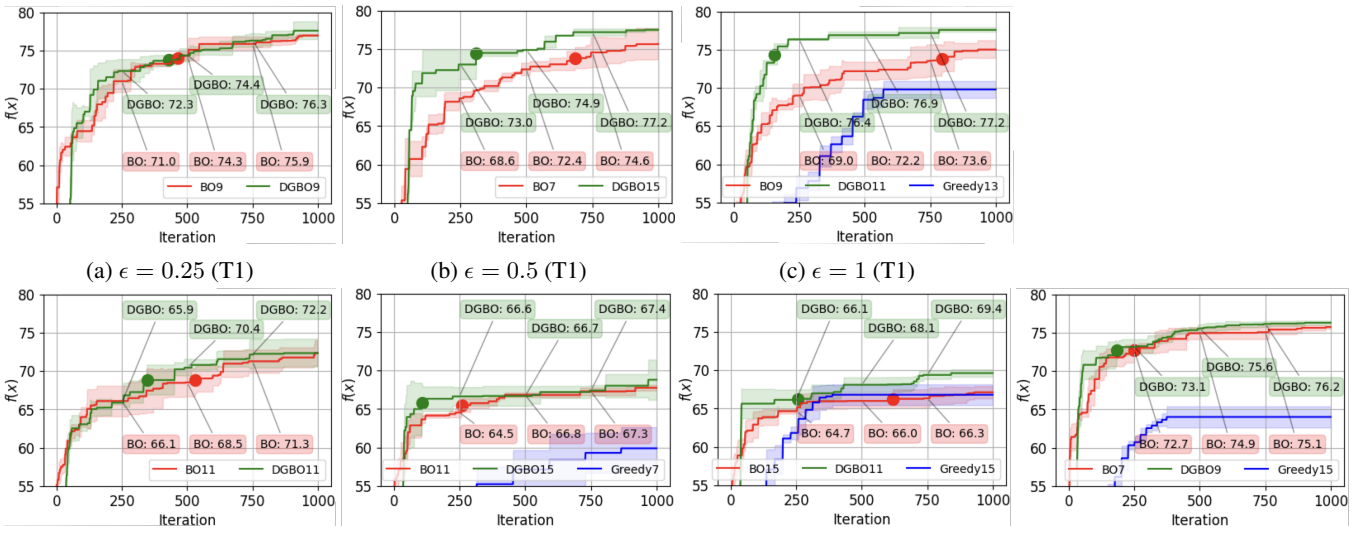
We use a publicly available, real-world dataset called Aruba (Cook et al. 2012). This dataset was collected from the home of an adult who lives alone but has regular visitors. The dataset contains 219 days worth of data generated by 31 motion sensors installed in that home. The following 11 ADLs were manually labeled: 1) meal preparation, 2) relax, 3) eating, 4) work, 5) sleeping, 6) wash dishes, 7) bed to toilet, 8) enter home, 9) leave home, 10) housekeeping, and 11) respirate. In this case, SIM_{sis} gets as input the real-world dataset and instead of simulating human activities and corresponding sensor triggers, it simply filters a subset of the sensor data based on the sensors present in the candidate sensor placement. This portion of the dataset is then used for activity recognition, with the first 70% of the days comprising the training set and the rest comprising the test data.

Results

We compare motion sensor placements found by DGBO (the proposed method), BO, GA, and greedy algorithm in our case studies. For implementation details, see the appendix in our preprint (Golestan, Ardakanian, and Boulanger 2023). For each value of $\epsilon\in\{0.25m, 0.5m, 1.0m\}$ in the simulation case study, we run each algorithm 5 times, using different seeds. Given the range of motion sensors in the simulator (a circle with radius of 1 meter), these ϵ values are reasonable because they allow some overlap between the areas covered

Method	Avg. F^1 ± one standard deviation (no. sensors used)						
	T1			T2			Aruba
	ϵ=0.25(m)	ϵ=0.5(m)	ϵ=1.0(m)	ϵ=0.25(m)	ϵ=0.5(m)	ϵ=1.0(m)	
GA	56.7±1.0 (9)	59.7±0.4 (11)	54.5±1.0 (9)	42.9±1.5 (10)	42.7±0.4 (6)	40.7±1.3 (5)	60.2±1.2 (7)
Greedy	—	—	69.7±1.1 (13)	—	59.8±2.7 (7)	66.8±1.3 (15)	64.0±1.3 (15)
BO	76.9±0.1 (9)	75.6±2.0 (7)	75.0±1.1 (9)	72.3±1.6 (11)	67.7±0.3 (11)	66.9±1.1 (13)	75.7±0.2 (9)
DGBO	77.6±1.1 (9)	77.5±0.1 (15)	77.6±0.2 (9)	72.3±1.8 (11)	68.7±2.6 (15)	69.6±0.4 (11)	76.3±0.2 (9)

Table 2: The performance of GA, Greedy, BO and DGBO in terms of the macro-averaged F^1 (Eqn. 2) in T1, T2 and Aruba. A dash indicates that no sensor placement was found after 1000 queries. We report the median number of sensors for GA.



(a) $\epsilon = 0.25$ (T1) (b) $\epsilon = 0.5$ (T1) (c) $\epsilon = 1$ (T1)

(d) $\epsilon = 0.25$ (T2) (e) $\epsilon = 0.5$ (T2) (f) $\epsilon = 1$ (T2) (g) Aruba (real-world dataset)

Figure 4: The average performance of DGBO, BO, and greedy (when available) for the best number of sensors found by each method versus the number of $f(x)$ queries.

by multiple sensors and maintaining a clear line of sight despite the obstacles that exist in the environment. For greedy, BO, and DGBO, we repeat the process after setting the total number of sensors to 5, 7, 9, 11, 13, and 15. GA decides the number of sensors automatically. Each algorithm can query the black-box function 1,000 times. Table 2 shows the performance (Eqn. 2) of DGBO, BO, greedy, and GA for the number of sensors that led to the best performance in each case (mentioned in brackets). In both T1 and T2, the greedy algorithm mostly exhausts the 1,000 black-box function queries for ϵ=0.25 and 0.5, failing to find a solution. For example, when ϵ=0.5 in T1, 1115 function queries would be needed to place 5 sensors.

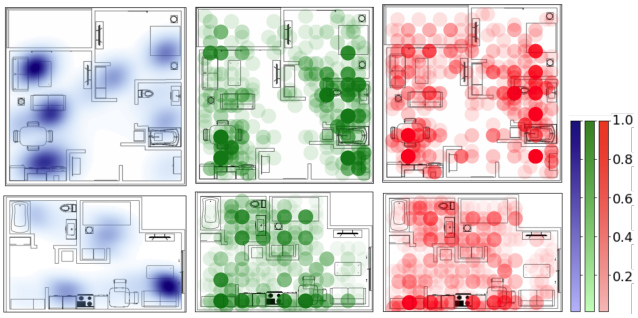
We make the following observations: 1) Both DGBO and BO significantly outperform GA and greedy for all ϵ values (a two-tailed t-test determines a significant difference (p<0.05)). This confirms our hypothesis that the surrogate model of $f(x)$ contains useful information in this context. 2) DGBO achieves better performance than BO in most cases. 3) Greedy algorithm performs well in T2 with ϵ=1. We attribute this to the rather small search space for this value of ϵ, increasing the chance of reaching the global optimum. 4) DGBO performs robustly across all testbeds, as evidenced by its convergence to a similar level of performance compared to the other methods. We believe this is because the information profiles guide DGBO towards more interesting parts of the indoor environment (see Fig. 1).

We investigate the fourth observation in detail. First, Fig. 4 depicts the average performance of DGBO, BO and greedy across the 5 runs after each iteration. We specifically focus on the best-performing number of sensors for each algorithm. We witness that DGBO quickly finds a high-quality sensor placement in all testbeds. To compare the sample efficiency of DGBO and BO, we find the first iteration at which DGBO and BO reach the 95% confidence interval (CI) of the best performance of DGBO after 1,000 iterations; these iterations are marked with green and red dots in Fig. 4, respectively. Table 3 shows how many fewer queries are required by DGBO to attain the same performance as BO, as the percentage of the number of queries executed by BO. On average, DGBO requires 55.4%, 58.9%, and 39% fewer queries than BO in T1, T2, and Aruba, respectively. This is a significant improvement, especially because these queries are typically expensive.

Second, Fig. 5 shows the best sensor locations found after 1,000 iterations in all runs of DGBO and BO (considering different ϵ values, random seeds, and target number of sensors) in T1 and T2. It also shows the spatial distribution of activities using a heatmap overlaid on the floor plan of each building, with dark/light blue showing more/less activities in

Testbed		$100\times$ 	avg.
T1	ϵ=**0.25**(**m**)	-17.9%	-55.4%
	ϵ=**0.5**(**m**)	-61.8%	
	ϵ=**1**(**m**)	-86.6%	
T2	ϵ=**0.25**(**m**)	-41.0%	-58.9%
	ϵ=**0.5**(**m**)	-71.7%	
	ϵ=**1**(**m**)	-64.1%	
Aruba		-39.6%	-39.6%

Table 3: The convergence analysis of DGBO compared to BO across our case studies. The avg(.) calculates the average value of red/green dots of each testbed for all ϵ values.



(a) (b) (c)

Figure 5: Illustration of the spatial distribution of activities (a), and the best sensor locations found in all runs of DGBO (b) and in all runs of BO (c). The first row shows T1 and the second row shows T2. The intensity of the color shows the likelihood of installing a sensor at that location.

the space. It can be readily seen that both methods place sensors in highly occupied areas, such as the kitchen and dining room. Yet, DGBO's sensor placements are more promising. Specifically, in T1, BO places sensors in areas where no activities were performed e.g., the balcony, entryway, left side of the bedroom, and right side of the living room, However, DGBO is less inclined to place sensors in these areas. The same argument can be made for T2.

Discussion

We have found that DGBO and BO outperform the conventional methods for finding motion sensor placements, i.e., genetic and greedy algorithms, resulting in significantly higher activity recognition accuracy. The DGBO learns the spatial distribution of activities during the search process and utilizes this distribution to consistently find high-quality sensor placements using noticeably fewer queries than BO. We wish to emphasize that in the BO literature, $f(x)$ queries are typically costly (Frazier 2018). In our problem, resolving these queries even through simulation is time-consuming, taking roughly 2 minutes on a computer with an Intel i7 4.00Ghz CPU and 16GB memory. Thus, coming up with sample efficient algorithms is key.

We argue that our spatial credit assignment strategy (Eqn. 8) leads to an accurate estimation of the expected information gain of each region after a small number of iterations. To show this, we plot the expected information gain

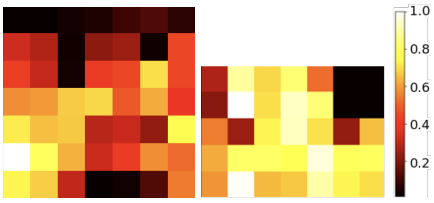


Figure 6: Expected information gain: T1 (left), T2 (right).

at iteration n=50 in Fig. 6. Comparing this to Fig. 5 (a), we conclude that our estimation of expected information gain becomes accurate early in the optimization process.

We wish to clarify that our framework in Fig. 2 is not designed for a particular choice of the activity classifier. To show this, we have repeated the process in T1 with ϵ=1 using two different classifiers: gradient boosting and K-nearest neighbor (KNN) with k=5. In both cases, the superior performance of BO and DGBO remains statistically significant and DGBO remains significantly more sample-efficient.

Further studies should be conducted using DGBO. A future work direction is to apply transfer learning to the expected information gain and use this information in other environments. Another future direction is to assess the effectiveness of DGBO in other emerging applications such as air pollution monitoring (Hellan, Lucas, and Goddard 2022), wildfire monitoring (Gholami et al. 2021), and effective emergency response (Ghosh and Varakantham 2018). Finally, we have explored the optimal placement of a sensor that is non-intrusive, inexpensive, and commonly used for activity detection in aging-in-place settings. Our method can be extended to other sensor types, e.g., mmWave radars.

While wearable sensors excel in activity detection, they must be worn at all times and charged regularly. This is not convenient for older people who may have cognitive impairment. Also, they collect sensitive data and tend to be more intrusive than motion sensors. Privacy protection is indeed crucial in aging-in-place settings. Our activity recognition technique classifies only the activities that are important for caregiving. Additionally, sensitive data will be securely shared with authorized caregivers in real-world applications.

Conclusion

This paper casts optimal sensor placement in indoor environments as a black-box optimization problem solved using Bayesian optimization. We introduce Distribution-Guided Bayesian Optimization, which incorporates the learned spatial distribution of activities into the acquisition function of Bayesian optimization. Our approach entails using (a) a high-fidelity simulator for modeling the indoor environment, its occupants, and sensors; (b) an optimization algorithm to find a sensor placement that maximizes the detection accuracy of ADL. We hypothesized that DGBO could explore the search space more effectively. To test this hypothesis, we evaluated DGBO in two simulated suites of an assisted living facility and using one real-world dataset where subjects performed ADLs. We compared the performance of DGBO, with BO and two widely-used baselines, i.e., genetic and greedy algorithms. Our result confirmed that DGBO finds high-quality sensor placements at a significantly lower cost.

References

Andersson, T. R.; Bruinsma, W. P.; Markou, S.; Requeima, J.; Coca-Castro, A.; Vaughan, A.; Ellis, A.-L.; Lazzara, M. A.; Jones, D.; Hosking, S.; et al. 2023. Environmental sensor placement with convolutional Gaussian neural processes. *Environmental Data Science*, 2: e32.

Astudillo, R.; and Frazier, P. I. 2021. Thinking inside the box: A tutorial on grey-box Bayesian optimization. In *2021 Winter Simulation Conference (WSC)*, 1–15. IEEE.

Barry, J.; and Thron, C. 2019. A computational physics-based algorithm for target coverage problems. In *Advances in nature-inspired computing and applications*, 269–290. Springer.

Briscoe, J.; Gebremedhin, A.; Holder, L. B.; and Cook, D. J. 2022. Adversarial Creation of a Smart Home Testbed for Novelty Detection. In *AAAI Spring Symposium on Designing AI for Open Worlds*.

Centers for Disease Control and Prevention. 2020. Older Adult Fall Prevention. https://www.cdc.gov/falls/data/index.html. Accessed: 2023-11-04.

Cook, D. J.; Crandall, A. S.; Thomas, B. L.; and Krishnan, N. C. 2012. CASAS: A smart home in a box. *Computer*, 46(7): 62–69.

Dahmen, J.; and Cook, D. 2019. SynSys: A synthetic data generation system for healthcare applications. *Sensors*, 19(5): 1181.

Deshwal, A.; Ament, S.; Balandat, M.; Bakshy, E.; Doppa, J. R.; and Eriksson, D. 2023. Bayesian Optimization over High-Dimensional Combinatorial Spaces via Dictionary-based Embeddings. In *International Conference on Artificial Intelligence and Statistics*, 7021–7039. PMLR.

Deshwal, A.; Belakaria, S.; Doppa, J. R.; and Fern, A. 2020. Optimizing discrete spaces via expensive evaluations: A learning to search framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3773–3780.

Deshwal, A.; Belakaria, S.; Doppa, J. R.; and Kim, D. H. 2022. Bayesian optimization over permutation spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6515–6523.

Fanti, M. P.; Roccotelli, M.; Faraut, G.; and Lesage, J.-J. 2017. Smart placement of motion sensors in a home environment. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 894–899. IEEE.

Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Freedman, R. G.; and Zilberstein, S. 2019. A unifying perspective of plan, activity, and intent recognition. In *Proceedings of the AAAI Workshops: Plan, Activity, Intent Recognition (Honolulu, HI)*, 1–8.

Gholami, S.; Kodandapani, N.; Wang, J.; and Ferres, J. L. 2021. Where there's Smoke, there's Fire: Wildfire Risk Predictive Modeling via Historical Climate Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15309–15315.

Ghosh, S.; and Varakantham, P. 2018. Dispatch guided allocation optimization for effective emergency response. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Golestan, S. 2020. SIM SIS Simulator. https://github.com/shadangolestan/SIM_SIS-Simulator. Accessed: 2022-12-20.

Golestan, S.; Ardakanian, O.; and Boulanger, P. 2023. Grey-box Bayesian Optimization for Sensor Placement in Assisted Living Environments. *arXiv preprint arXiv:2309.05784*.

Golestan, S.; Nikolaidis, I.; and Stroulia, E. 2020. Towards a Simulation Framework for Smart Indoor Spaces. *Sensors*, 20(24): 7137.

Guestrin, C.; Krause, A.; and Singh, A. P. 2005. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, 265–272.

Gungor, O.; Rosing, T. S.; and Aksanli, B. 2020. RESPIRE: Robust Sensor Placement Optimization in Probabilistic Environments. In *2020 IEEE Sensors*, 1–4. IEEE.

Hellan, S. P.; Lucas, C. G.; and Goddard, N. H. 2022. Bayesian Optimisation for Active Monitoring of Air Pollution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 11908–11916.

Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2011. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, 507–523. Springer.

Li, Y.; Shen, Y.; Zhang, W.; Chen, Y.; Jiang, H.; Liu, M.; Jiang, J.; Gao, J.; Wu, W.; Yang, Z.; et al. 2021. Openbox: A generalized black-box optimization service. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3209–3219.

Lifestyle Options. 2023. Lifestyle Options. https://lifestyleoptions.ca/terra-losa/. Accessed: 2022-12-20.

Mori, N.; Takeda, M.; and Matsumoto, K. 2005. A comparison study between genetic algorithms and bayesian optimize algorithms by novel indices. In *Proceedings of the 7th annual conference on genetic and evolutionary computation*, 1485–1492.

Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and De Freitas, N. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175.

Sutton, R. S. 1984. *Temporal credit assignment in reinforcement learning*. University of Massachusetts Amherst.

Thomas, B. L.; Crandall, A. S.; and Cook, D. J. 2016. A Genetic Algorithm approach to motion sensor placement in smart environments. *Journal of Reliable Intelligent Environments*, 2(1): 3–16.

Vlasenko, I.; Nikolaidis, I.; and Stroulia, E. 2014. The smart-condo: Optimizing sensor placement for indoor localization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3): 436–453.

Weissteiner, J.; Heiss, J.; Siems, J.; and Seuken, S. 2023. Bayesian optimization-based combinatorial assignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 5858–5866.

Wu, H.; Liu, Z.; Hu, J.; and Yin, W. 2020. Sensor placement optimization for critical-grid coverage problem of indoor positioning. *International Journal of Distributed Sensor Networks*, 16(12): 1550147720979922.

Yang, C. 2021. An adaptive sensor placement algorithm for structural health monitoring based on multi-objective iterative optimization using weight factor updating. *Mechanical Systems and Signal Processing*, 151: 107363.

Yu, X.; Ergun, K.; Cherkasova, L.; and Rosing, T. Š. 2020. Optimizing sensor deployment and maintenance costs for large-scale environmental monitoring. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11): 3918–3930.