

# Selection of Sensors for Efficient Transmitter Localization

Arani Bhattacharya\*, Caitao Zhan<sup>†</sup>, Abhishek Maji\*, Himanshu Gupta<sup>†</sup>, Samir R. Das<sup>†</sup>, Petar M. Djuric<sup>†</sup>

\*KTH Royal Institute of Technology, <sup>†</sup>Stony Brook University

\*{aranib, amaji}@kth.se, <sup>†</sup>{cbzhan, hgupta, samir}@cs.stonybrook.edu, petar.djuric@stonybrook.edu

**Abstract**—We address the problem of localizing an (illegal) transmitter using a distributed set of sensors. Our focus is on developing techniques that perform the transmitter localization in an efficient manner, wherein the efficiency is defined in terms of the number of sensors used to localize. Localization of illegal transmitters is an important problem which arises in many important applications, e.g., in patrolling of shared spectrum systems for any unauthorized users. Localization of transmitters is generally done based on observations from a deployed set of sensors with limited resources, thus it is imperative to design techniques that minimize the sensors’ energy resources.

In this paper, we design greedy approximation algorithms for the optimization problem of selecting a given number of sensors in order to maximize an appropriately defined objective function of localization accuracy. The obvious greedy algorithm delivers a constant-factor approximation only for the special case of two hypotheses (potential locations). For the general case of multiple hypotheses, we design a greedy algorithm based on an appropriate auxiliary objective function—and show that it delivers a provably approximate solution for the general case. We develop techniques to significantly reduce the time complexity of the designed algorithms, by incorporating certain observations and reasonable assumptions. We evaluate our techniques over multiple simulation platforms, including an indoor as well as an outdoor testbed, and demonstrate the effectiveness of our designed techniques—our techniques easily outperform prior and other approaches by up to 50-60% in large-scale simulations.

## I. INTRODUCTION

Wireless transmitter localization via analysis of the received signal from multiple receivers or sensors is an important problem. While the problem has been widely explored, the problem exposes new challenges in many emerging applications due to the constraints of the application. In this work, we are specifically interested in a distributed monitoring system where a set of distributed RF sensors are tasked to detect and localize transmitters. These transmitters could be of various type. For example, in certain spectrum allocation scenarios, unknown primary transmitters need to be detected/localized. Or, in spectrum patrolling scenarios, unauthorized transmitters need to be detected/localized [1]. Recent work has explored new approaches for such monitoring where the RF sensors are crowdsourced, perhaps using various low-cost spectrum sensing platforms [2], [3]. The crowdsourcing deploys a large number of sensors. Fine grained spectrum sensing is implemented by creating suitable incentive mechanisms [4], [2].

Crowdsourcing makes the sensing *cost-conscious*. The cost here could be incentivization cost, cost of power, backhaul bandwidth on the part of the spectrum owner or the opportunity cost – being low-cost platform, the sensors may be able to

only sense smaller spectrum bands at a time. Thus, involving only a small number of sensors or sensors with low overall cost budget (for a suitable cost model) for sufficiently accurate localization performance is critical. Prior work [2] that discuss sensor selection in this context only presents heuristics without any performance guarantees.

We do not use geometric approaches which rely on hard-to-model mapping of received power to distance. Instead, we use a hypothesis-driven, Bayesian approach for localization [5]. We focus on the optimization problem of selecting a certain number of sensors from among the deployed sensors such that an appropriately defined objective of localization accuracy is maximized. This optimization problem can also be used to solve the dual problem of selecting a minimum number of sensors (or sensors with the minimum total cost budget) to ensure at least a given localization accuracy. We adopt the framework of a hypothesis-driven localization approach wherein each hypothesis represents a configuration (location, power, etc.) of the potential transmitters and then the localization is equivalent to determining the most-likely prevailing hypothesis. See Figure 1. The hypothesis-driven framework does not require an assumption of a propagation model, and works for arbitrary signal propagation characteristics. The framework does, however, require prior training to build joint probability distributions of observation vectors for each hypothesis.

**Our Contributions.** In the above hypothesis-based framework, we develop an overall approach that enables selection of sensors that are most relevant to localize transmitters. In particular, we develop algorithms that aim to maximize localization accuracy for a given budget of number of sensors to be used for localization. More specifically, we make the following contributions in the paper.

- 1) We design a greedy algorithm (GA) that selects sensors iteratively to maximize the objective function of localization accuracy, under the constraint of number of sensors selected. We prove that GA yields a constant-factor approximate solution for the special case of the problem wherein there are only two hypotheses.
- 2) For the general case of more than two hypotheses, we design an alternate greedy scheme (called AGA) based on maximizing an auxiliary objective function. We prove that AGA delivers a solution that has (i) an auxiliary objective value within a constant factor of the optimal auxiliary objective value, as well as (ii) a localization error within

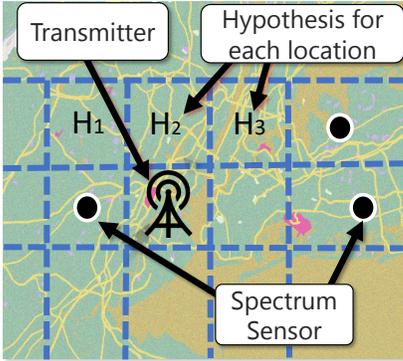


Fig. 1: Hypothesis-driven localization. The figure shows the simple case of localizing a single transmitter with fixed power; thus, there is a hypothesis created for each potential location. Observations from deployed sensors are analyzed to determine the most likely prevailing hypothesis (and thus, location).

a certain factor of the optimal localization error.

- 3) We optimize the time complexity of our developed algorithms by a substantial factor, based on certain observations and reasonable assumptions. In addition, we generalize our techniques to more practical and useful settings.
- 4) We evaluate the performance of the developed algorithms over multiple evaluation platforms: (1) large-scale simulation using synthetically generated data using established signal propagation models, and (2) publicly available experimental data trace collected over an indoor WiFi network with 44 sensors, and (3) our own data collection using 18 outdoor software radio sensors in the 915 MHz band with a custom transmitter. Results show that our techniques outperform other state-of-the-art algorithm [2] substantially (up to a factor of 50-60%).

A preliminary version of this paper has been accepted for publication at IEEE Infocom 2020 [6]. This version of the paper describes additional results about the performance of the algorithm, as well as additional details about the experiments. It also contains proofs of multiple lemmas and theorems that had been omitted from the preliminary version.

## II. BACKGROUND AND MOTIVATION

**Problem Setting.** The overall setting of the transmitter localization problem is as follows. Consider a geographic area, with a number of spectrum sensors deployed or available (if attached to mobile devices) at known locations. At any instant, one or more transmitters are allowed to transmit signals (on a common frequency). Each deployed/available spectrum sensor senses and processes the aggregate received signal, and reports appropriate metric (i.e., total received power or signal strength) to a central server which estimates the location of the transmitter(s) using the maximum-likelihood hypothesis algorithm as described below. The overall objective of our paper is to develop techniques to select an optimal subset of sensors in order to accurately localize any present transmitters. Though our developed techniques naturally extend to the case of multiple transmitters, for simplicity, we implicitly assume at most a single transmitter present at any instant. We consider

the extension to multiple transmitters in §III-G. We start with defining basic notations used throughout the paper.

**Hypotheses, Observations, and Inputs.** We discretize the given space into locations  $l_1, l_2, \dots$ , and transmit power of a potential transmitter is similarly discretized into levels  $p_1, p_2, \dots$ . We represent potential “configurations” of the possible transmitter by hypotheses  $H_0, H_1, \dots, H_m$ , where each hypothesis  $H_i$  represents a configuration  $(l_i, p_i)$  of location  $l_i$  and transmit power  $p_i$  of a potential transmitter (see Figure 1). We use the convention that hypothesis  $H_0$  corresponds to no transmitter being present. Localizing any potential transmitter is thus equivalent to determining the prevailing hypothesis. To do this, we use observations from a set of deployed sensors. We denote the observation vector of a subset of sensors  $\mathbf{T}$  by  $\mathbf{x}_{\mathbf{T}}$  (we usually drop the subscript  $\mathbf{T}$ , as it is clear from the context).

**Inputs.** For a given set of sensors deployed over an area, we assume the following available inputs, obtained via a priori training, data gathering and/or analysis<sup>1</sup>:

- Prior probabilities of the hypotheses, i.e.  $P(H_i)$ , for each hypothesis  $H_i$ .
- Joint probability distribution (JPD) of sensors’ observations for each hypothesis. More formally, for each hypothesis  $H_j$ , we assume  $P(\mathbf{x}_{\mathbf{S}}|H_j)$  to be known for each observation  $\mathbf{x}_{\mathbf{S}}$  for the entire set  $\mathbf{S}$  of deployed sensor. Note that this also gives us the JPD’s of each subset  $\mathbf{T} \subseteq \mathbf{S}$ .

**Maximum a Posteriori Localization (MAP) Algorithm.** We use Bayes rule to compute the likelihood probability of each hypothesis, from a given observation vector  $\mathbf{x}_{\mathbf{T}}$  for a subset of sensors  $\mathbf{T}$ :

$$P(H_i|\mathbf{x}_{\mathbf{T}}) = \frac{P(\mathbf{x}_{\mathbf{T}}|H_i)P(H_i)}{\sum_{j=0}^m P(\mathbf{x}_{\mathbf{T}}|H_j)P(H_j)} \quad (1)$$

We select the hypothesis that has the highest probability, for given observations of a set of sensors. Formally, the MAP algorithm returns the hypotheses based on the following equation:

$$\arg \max_{i=0}^m P(H_i|\mathbf{x}_{\mathbf{T}}) \quad (2)$$

The above MAP algorithm to determine the prevailing hypothesis is known to be *optimal* [8], i.e., it yields minimum probability of (misclassification) error. The above hypothesis-based approach to localization works for arbitrary signal propagation characteristics, and in particular, obviates the need to assume a propagation model. However, it does incur a one-time training cost to obtain the JPDs, which can be optimized via independent techniques [9].

**Selection of Sensors for Localization.** As mentioned above, in a typical setting, spectrum sensors may be deployed at pre-determined locations or available at certain locations (if part of mobile devices) to sense unauthorized signals and thus localize any unauthorized transmitters. Two immediate problems of interest in this context are: where to deploy given a number of

<sup>1</sup>In our concurrent work [7], we discuss novel interpolation techniques to minimize such training cost.

sensors, and once deployed/available, which subset of sensors to select for localization. The latter problem of selection of sensors is motivated by the fact that, in most realistic settings, the sensors (or their mobile devices) are not tethered to AC power outlets and hence have limited energy resources. Moreover, spectrum sensors also incur cost in transmitting sensing data to the fusion/cloud center [10]. Thus, it is critical to optimize resources and costs incurred in localization of unauthorized transmitters, e.g., via the selection of an optimal set of sensors. Note that the sensor-selection problem can also be used to effectively *deploy* a given number of sensor, by assuming sensors available at all potential locations.

### III. OPTIMAL SENSOR SELECTION FOR INTRUDER LOCALIZATION

In this section, we address the problem of sensor selection for transmitter localization; informally, the problem is to select an optimal set of  $B$  sensors such that the overall probability of error of localizing a transmitter is minimized, given appropriate JPDs as discussed in the previous section. We start with formulating the problem in the following subsection. In following subsection, we present a greedy algorithm for it and prove that it is guaranteed to deliver an approximation solution for the special case of two hypotheses. However, as shown, the greedy algorithm can perform arbitrarily bad for the general case of multiple hypotheses. Thus, we then modify our algorithm to use an “auxiliary” objective function and show that the modified algorithm delivers an approximation solution for the general case of multiple hypotheses albeit with a slightly worse approximation ratio. Finally, we discuss optimizing the computation complexity of the designed algorithms, certain extensions and other issues.

#### A. LSS Problem Formulation

We start with formally defining the optimization objective (probability of error or misclassification) for a given subset of sensors. Then, we formally define the sensor selection problem, hereto referred to as *Localization Sensor Selection (LSS)* problem. Throughout this section, we use hypotheses  $H_0$  to represent the hypotheses with no transmitters present, and  $H_i$  to represent the hypotheses wherein a transmitter is present in  $i^{\text{th}}$  configuration.

**Probability of Error ( $P_{\text{err}}(\mathbf{T})$ ).** Recall that, for a given observation vector, the MAP localization algorithm outputs the hypothesis that has the most likelihood among the given hypotheses. Thus, MAP can also be looked upon as a classification technique. Given a subset of sensors  $\mathbf{T}$ , we define the *probability of error* or misclassification as the probability of the MAP algorithm outputting a hypothesis different from the actual *ground truth* (i.e., prevailing hypothesis). The expected or overall probability of error is an expectation of the probability of error over all possible prevailing hypotheses and/or observation vectors  $\mathbf{x}_{\mathbf{T}}$  from  $\mathbf{T}$ . Our techniques generalize to the notion of distance-based localization error, as discussed in §III-G.

Formally, let  $\text{MAP}(\mathbf{x})$  be the output of the MAP algorithm on observation vector  $\mathbf{x}$  from a given subset of sensors  $\mathbf{T}$ . Given

$H_i$  as the ground truth and  $\mathbf{x}$  as the observation vector, the probability of error  $P_{\text{err}}(\mathbf{T}|H_i, \mathbf{x})$  can be written as:

$$P_{\text{err}}(\mathbf{T}|H_i, \mathbf{x}) = \mathbf{1}[\text{MAP}(\mathbf{x}) \neq i|H_i], \quad (3)$$

where  $\mathbf{1}$  is an indicator function which is equal to 1 if the predicate is true, and 0 otherwise. Since expectation over the data point of an indicator function is its probability, we take the expectation over  $\mathbf{x}$  on both sides to get:

$$P_{\text{err}}(\mathbf{T}|H_i) = P(\text{MAP}(\mathbf{x}) \neq i|H_i) \quad (4)$$

Above, the probability is over the random variable  $\mathbf{x}$ . Now, if the ground truth hypothesis is also not given, we can compute an expectation over all possible hypotheses. Thus, the (overall) *probability of error* for a given set of sensors  $\mathbf{T}$  is given by:

$$P_{\text{err}}(\mathbf{T}) = \sum_{i=0}^{m-1} P(\text{MAP}(\mathbf{x}) \neq i|H_i)P(H_i) \quad (5)$$

**Localization Accuracy Function,  $O_{\text{acc}}(\mathbf{T})$ .** To facilitate a greedy approximation solution, we formulate our sensor selection as a maximization problem—and thus, define a corresponding maximization objective. In particular, we define the *localization accuracy*  $O_{\text{acc}}(\mathbf{T})$  as  $1 - P_{\text{err}}(\mathbf{T})$ . Based on the above equation Eqn. 5, we get the expression for  $O_{\text{acc}}(\mathbf{T})$  as:

$$O_{\text{acc}}(\mathbf{T}) = 1 - P_{\text{err}}(\mathbf{T}) = \sum_{i=0}^{m-1} P(\text{MAP}(\mathbf{x}) = i|H_i)P(H_i) \quad (6)$$

**Localization Sensor Selection (LSS) Problem.** Consider a geographic area with a set of sensors  $\mathbf{S}$  deployed. Given a set of hypotheses and JPD’s, as defined in previous section, the OSS problem is to select a subset  $\mathbf{T} \subseteq \mathbf{S}$  of sensors with minimum probability of error  $P_{\text{err}}(\mathbf{T})$  (or maximum localization accuracy  $O_{\text{acc}}(\mathbf{T})$ ), under the constraint that  $|\mathbf{T}|$  is at most a given budget  $B$ . The above formulation implicitly assumes a uniform cost for each sensor; we generalize our techniques to handle non-uniform sensor costs (see §III-G).

We show that the above LSS problem is NP-hard, via reduction from the well-known maximum-coverage problem (Appendix A). Thus, we develop approximation algorithms below; in particular, our focus is on developing greedy approximation algorithms. The key challenge lies in showing that the objective function satisfies certain desired properties that ensure the approximability of the algorithm.

#### B. Transmitter and Sensor Model

We now formally define the assumptions that would allow us to ensure the approximability of our algorithm. First, we assume that the joint probability distribution (JPD) of the distributions follow a joint Gaussian distribution with the means  $(\mathbf{p}_i, \Sigma)$  for all hypotheses  $H_i \forall i = 0, \dots, m-1$ . Note that the covariance matrix remains same across hypotheses, since the correlation and noise are properties of the sensors. The means  $\mathbf{p}_i$  can be different, as different power values are received by the sensors depending on the location of the transmitter.

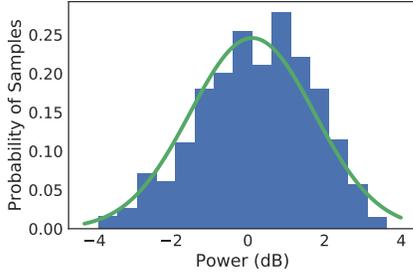


Fig. 2: Distribution of the received power from a transmitter at an RTL-SDR sensor, and the Gaussian fit (green line) of the observed distribution.

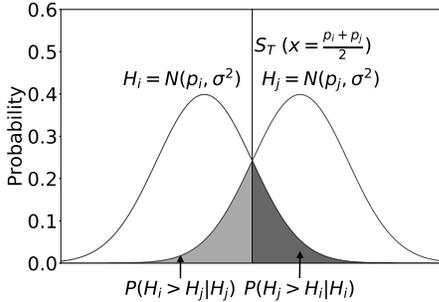


Fig. 3: Classification of a data point between two Gaussians using a threshold.

### C. Properties of MAP Algorithm

To explain our selection algorithm, we first need to explain a few properties of MAP algorithm. Assume that there are two hypotheses  $H_i$  and  $H_j$ , with distributions  $(p_i, \sigma^2)$  and  $(p_j, \sigma^2)$  as well as priors  $P(H_i)$  and  $P(H_j)$  respectively, where  $p_i, p_j \in \mathbb{R}$ . Without loss of generality, we assume that  $p_i < p_j$ . In this case, given a data point  $X$ , the MAP algorithm works by comparing it with a fixed threshold  $S_T$  (shown in Figure 3). If  $X \leq S_T$ , then MAP classifies  $X$  as  $H_i$ , i.e.  $\text{MAP}(X) = i$ , otherwise it classifies  $X$  as  $H_j$ , i.e.  $\text{MAP}(X) = j$ . Note that because this is a stochastic decision, there will always be some probability of classification error, depending on the value of  $S_T$ . The MAP algorithm uses the threshold value of  $S_T = \frac{p_i + p_j}{2}$ , and it is well-known that this value of  $S_T$  provides the lowest probability of classification error. Formally, we write this as:

$$X \underset{H_i}{\overset{H_j}{\gtrless}} \frac{p_i + p_j}{2} + \log \frac{P(H_i)}{P(H_j)} \quad (7)$$

We now explain the case for multidimensional distributions, where  $H_i$  and  $H_j$  are given by  $(\mathbf{p}_i, \Sigma)$  and  $(\mathbf{p}_j, \Sigma)$  respectively ( $\mathbf{p}_i, \mathbf{p}_j \in \mathbb{R}^n, \Sigma \in \mathbb{R}^{n \times n}$ ). In this case, the classification of a given data vector can be done by comparing with a hyperplane. However, this problem of classification between distributions with multiple dimensions can be *reduced* to classification between distributions with single dimensions, using the following theorem:

**Theorem 1.** *Given the hypotheses  $H_i \sim N(\mathbf{p}_i, \Sigma)$  and  $H_j \sim N(\mathbf{p}_j, \Sigma)$ , a data vector  $\mathbf{x} = [x_1 \dots x_n]$  can be classified by*

applying the following threshold test:

$$\mathbf{x}^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) \underset{H_i}{\overset{H_j}{\gtrless}} \frac{1}{2} (\mathbf{p}_i + \mathbf{p}_j)^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) \quad (8)$$

We prove this in Appendix B. We call the LHS of Eqn (8) as test statistic  $T(\mathbf{x})$ . We also show as a corollary of the theorem that the test statistic itself follows a Gaussian distribution of  $N(\mathbf{p}_i \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i), (\mathbf{p}_j - \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i))$  and  $N(\mathbf{p}_j \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i), (\mathbf{p}_j - \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i))$  if  $\mathbf{x}$  is from  $H_i$  and  $H_j$  respectively. Thus, our problem is now exactly equivalent to classification using MAP to classify a data point between two Gaussians with known means and same variance.

### D. Greedy Algorithm (GA)

In this subsection, we analyze a simple greedy approach and show that it delivers a constant-factor approximate solution for the special case of two hypotheses. In the next subsection, we present a modified greedy algorithm for the general case of more than two hypotheses.

**Greedy Algorithm (GA):** A straightforward algorithm for the LSS problem is a greedy approach wherein we iteratively select a single sensor at each stage. At each stage, we select the sensor that improves the localization accuracy  $O_{\text{acc}}(\mathbf{T})$  the most. The algorithm iterates until the given budget  $B$  is reached. We call this algorithm Greedy Algorithm (GA); see Algorithm 1 for the pseudo-code.

**Constant-Factor Approximation for 2 Hypotheses.** The approximation result of GA depends on two lemmas, which we prove in the appendix. The first lemma says that addition of a sensor to a given subset never reduces the value of  $O_{\text{acc}}$ :

**Lemma 1.** *The objective  $O_{\text{acc}}(\mathbf{T})$  is monotone in nature, i.e. if some sensor  $s_k \in \mathbf{S} \setminus \mathbf{T}$ , then  $O_{\text{acc}}(\mathbf{T} \cup \{s_k\}) \geq O_{\text{acc}}(\mathbf{T})$ .*

The second lemma says that the amount of increase in accuracy follows a law of diminishing returns:

**Lemma 2.** *The objective  $O_{\text{acc}} \mathbf{T}$  is submodular in nature, i.e. if some sensor  $s_k \in \mathbf{S} \setminus \mathbf{T}_2$ , where  $\forall \mathbf{T}_1 \subseteq \mathbf{T}_2 \subseteq \mathbf{S}$ , we have:*

$$O_{\text{acc}}(\mathbf{T}_1 \cup \{s_k\}) - O_{\text{acc}}(\mathbf{T}_1) \geq O_{\text{acc}}(\mathbf{T}_2 \cup \{s_k\}) - O_{\text{acc}}(\mathbf{T}_2) \quad (9)$$

Intuitively, these lemmas follow from Theorem 1, where we showed that the problem of identifying the right hypothesis is equivalent to classifying between two unidimensional Gaussians. It is well-known that if an objective is monotone and submodular, then GA gives an approximation result [11]. Thus, the following theorem on the performance of GA now holds:

**Theorem 2.** *For the special case of two hypotheses, GA gives a subset  $\mathbf{T}$  of sensors whose localization accuracy is at least 63% of the optimal.*  $\square$

**Performance of GA for more than two Hypotheses.** For the case of more than two hypotheses, GA no longer provides a constant-factor approximation. In fact, we can show via a counter-example that the  $O_{\text{acc}}()$  is not submodular for more than 2 hypotheses. We show this by providing a counter-example in Appendix C.

---

**Algorithm 1** Greedy Algorithm (GA).
 

---

**INPUT:** Set of available sensors  $\mathbf{S}$ , budget  $B$ , objective  $O_{\text{acc}}$   
**OUTPUT:** Subset of sensors  $\mathbf{T}$

```

1:  $\mathbf{T} \leftarrow \phi$ 
2: while  $|\mathbf{T}| \leq B$  do
3:    $L \leftarrow O_{\text{acc}}(\mathbf{T})$ 
4:    $\max \leftarrow 0$ 
5:   for all  $s \in \mathbf{S} \setminus \mathbf{T}$  do
6:      $M = O_{\text{acc}}(\mathbf{T} \cup \{s\}) - L$ 
7:     if  $M > \max$  then
8:        $\max \leftarrow M$ 
9:        $r \leftarrow s$ 
10:    end if
11:  end for
12:   $\mathbf{T} \leftarrow \mathbf{T} \cup \{r\}$ 
13: end while
14: return  $\mathbf{T}$ 

```

---

### E. Auxiliary Greedy Algorithm (AGA)

In the section, we design an approximation algorithm for the general case of multiple hypotheses based on an auxiliary objective function. To do so, we first analyze the proof of Theorem 2 and see why it doesn't generalize if the number of hypotheses is greater than 2. This insight helps in defining an "auxiliary" objective function that is the key to designing the approximation algorithm for the general case.

**Auxiliary Function.** Let us consider a special case of MAP algorithm, viz.,  $\text{MAP}_{ij}$  which compares the likelihood of only two hypothesis  $H_i$  and  $H_j$  and returns the one with a higher likelihood. It is easy to formulate the objective function  $O_{\text{acc}}$  in terms of  $\text{MAP}_{ij}$  too. From Equation 6, we easily get:

$$O_{\text{acc}}(\mathbf{T}) = \sum_{i=0}^{m-1} P\left(\bigcap_{j \neq i} \text{MAP}_{ij}(\mathbf{x}) = i | H_i\right) P(H_i) \quad (10)$$

$$O_{\text{acc}}(\mathbf{T}) = \sum_{i=0}^{m-1} [1 - P\left(\bigcup_{j \neq i} \text{MAP}_{ij}(\mathbf{x}) = j | H_i\right)] P(H_i) \quad (11)$$

Above,  $\mathbf{x}$  represents the observation vector for the set of sensors  $\mathbf{T}$ . For the case of two hypothesis, the above expression is just  $\sum_{i=0}^1 [1 - P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i)] P(H_i)$  where  $j$  is 1 if  $i$  is 0 and vice-versa; Theorem 2 essential shows that the term  $P(\text{MAP}_{ij}(\mathbf{x}) = i | H_i)$  is submodular. However, for the case of multiple hypothesis, computing the probability for a union of events involves product (and sum) of appropriate probability terms. Note that product of submodular functions need not be submodular, while sum of submodular functions is submodular. Thus, we *approximate* the above  $O_{\text{acc}}()$  expression as follows, so that it is a sum of submodular terms. In effect, in defining the auxiliary objective  $O_{\text{aux}}()$ , we estimate the probability of union of events in the above equation by just taking a summation of the probability of events, i.e., we ignore the other terms involving subsets of events. Formally, we define the auxiliary objective  $O_{\text{aux}}()$  for a set of sensors  $\mathbf{T}$  as:

$$O_{\text{aux}}(\mathbf{T}) = 1 - \sum_{i=0}^{m-1} \sum_{j \neq i} P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i) P(H_i) \quad (12)$$

The above auxiliary objection function is submodular if the JPDs are Gaussian, as it is a sum of submodular functions ( $P(\text{MAP}_{ij}(\mathbf{x}) = i | H_i)$  is submodular, as per Theorem 2's proof). Note that, for a competitive algorithm for the original LSS problem, we also need to show that maximizing  $O_{\text{aux}}()$  also maximizes the original objective function  $O_{\text{acc}}()$ .

**Auxiliary Greedy Algorithm (AGA).** We now modify our Greedy Algorithm (Algorithm 1) to iteratively maximize the auxiliary objective  $O_{\text{aux}}()$  instead of the original objective  $O_{\text{acc}}()$ . We call this algorithm as Auxiliary Greedy Algorithm (AGA). From the submodularity of the  $O_{\text{aux}}()$  for Gaussian JPDs, it is easy to see that AGA delivers a solution  $\mathbf{T}$  s.t.  $O_{\text{aux}}(\mathbf{T})$  is within 63% of the optimal  $O_{\text{aux}}()$  possible. The following lemma states that maximizing  $O_{\text{aux}}$  also maximizes  $O_{\text{acc}}$ . See Appendix E for a proof.

**Lemma 3.** *Let  $\mathbf{T}$  be a subset of sensors already selected by AGA at some iteration. We claim that  $O_{\text{aux}}(\mathbf{T}) \leq O_{\text{acc}}(\mathbf{T}) \leq 1 - \frac{1}{k}(1 - O_{\text{aux}}(\mathbf{T}))$ , where  $k$  is a value less than  $m$  that decreases as  $\mathbf{T}$  grows (i.e., over AGA's iterations).  $\square$*

We empirically evaluate the value of  $k$  defined above in §IV. The above lemma yields the below theorem, whose proof is shown in Appendix F.

**Theorem 3.** *For Gaussian JPDs, AGA delivers a subset  $\mathbf{T}$  of sensors such that*

$$P_{\text{err}}(\mathbf{T}) \leq 0.37 + 0.63kP_{\text{err}}(\text{OPT}),$$

where  $k$  is as defined in the above Lemma and  $\text{OPT}$  is the optimal solution.  $\square$

### F. Optimizing AGA's Computation Cost

In a straightforward implementation of AGA (akin to Algorithm 1 for GA),  $O_{\text{aux}}$  function is computed (using Eqn. (12))  $Bn$  number of times where  $n$  is the total number of sensors. Eqn. (12) requires  $m^2$  computations of the expectation  $P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i)$ , which, for Gaussian distributions, effectively requires computing the formula shown in Eqn. (11) of auxiliary material, and thus takes  $O(B^2)$  time as it involves matrix multiplication of the observation vector of dimension  $B$  with the covariance matrix of dimension  $B \times B$ . Thus, the overall time complexity of a straightforward implementation of AGA is  $O(m^2 n B^3)$ . As mentioned before (and in §II), the number of hypotheses  $m$  can be large due to the large number of potential transmitter locations and power values; however, we can reduce the time complexity to  $O(Bn)$  as discussed below, based on some observations and optimizations.

**Reducing Number of Comparisons.** Consider a sensor  $s$  whose benefit is to be computed in the `FOR` loop of Algorithm 1. Below, we show that effectively we only need to consider a constant number of  $(H_i, H_j)$  pairs in Eqn. (12) when computing  $s$ 's benefit, and thus removing the  $m^2$  factor from the time complexity. We implicitly assume a single transmitter in the below discussion, and later extend our argument to multiple transmitters. Let us use  $R$  to denote the maximum transmission "range" of the transmitter; formally,  $R$  is such that, beyond  $R$ , the probability distribution of the

signal received from the transmitter is similar to the signal received when there is no transmitter present. We stipulate that any observation  $x_s$  at  $s$ ,  $P(x_s|H_{i1}) = P(x_s|H_{i2})$  for any two hypotheses  $H_{i1}$  and  $H_{i2}$  whose corresponding locations  $l_{i1}$  and  $l_{i2}$  are more than  $R$  distance away from  $s$ . The implication of the above observation is that, for a given sensor  $s$ , we can group all the hypotheses  $H_i$  with corresponding location  $l_i$  more than  $R$  distance away from  $s$  into one single “super” hypothesis  $H_s$ . Then, if the total number of hypotheses with corresponding locations within a distance of  $R$  from  $s$  is say  $G_R$ , then the total number of comparisons between pairs of hypotheses in Eqn. (12) is effectively only  $(G_R + 2)^2$ , involving  $G_R$  hypotheses,  $H_0$ , and  $H_s$ . The above brings down the overall time complexity of AGA to  $O(G_R^2 n B^3)$ . Note that  $G_R$  is essentially equal to the number of grid locations within a circle of radius  $R$  times the total number of power levels, and thus, can be considered as constant (does not vary across problem instances)—which reduces AGA’s time complexity to  $O(nB^3)$ .

**Independent Sensor Observations.** If we assume that the observations across sensors are conditionally independent, the JPDs can be instead represented by independent probability distributions at each sensor location. In this case, the covariance matrix is purely diagonal, which allows us to “incrementally” compute the benefit of a sensor from one AGA iteration to another and thus reduce AGA’s time complexity by an additional factor of  $B^2$ —and thus to  $O(nB)$ . See Appendix G for details.

### G. Generalizations

**Weighted (Distance-Based) Objective Function.** The probability of error  $P_{\text{err}}$  function penalizes *uniformly* for each misclassification. However, in general, it would be useful to assign different penalties or weights for different misclassifications. E.g., Eqn (12) should be generalized to:

$$O'_{\text{aux}}(\mathbf{T}) = 1 - \sum_{i=0}^m \sum_{j \neq i} w_{ij} P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i) P(H_i)$$

Above,  $w_{ij}$  is the weight function. We note that our techniques and proofs of performance guarantees generalize easily to the above generalized function, irrespective of the weight function. In particular, weight  $w_{ij}$  can be the Euclidean distance between the locations  $l_i$  and  $l_j$  corresponding to the hypotheses  $H_i$  and  $H_j$ . For the general case of multiple transmitters, where  $H_i$  and  $H_j$  may represent configuration of multiple transmitters, a minimum-cost matching based objective can be used to define the weight  $w_{ij}$ ; if the number of transmitters in  $H_i$  and  $H_j$  are different, then an appropriately penalty for misses or false-alarms can be added to the weight.

**Non-Uniform Sensor Cost.** Another generalization of interest is to allow non-uniform cost for sensors, e.g., to prefer sensors with more (remaining) battery resources. Here, each sensor  $s$  may have a different cost  $c(s)$ , and the LSS problem constraint becomes: total cost of the selected set of sensors must be less than a given cost budget. For this version of the LSS problem, our algorithms need to be slightly modified in that

we should pick the sensor that offers the highest improvement in the objective function per unit cost. To ensure a theoretical performance guarantee, we also need to use the “knapsack trick,” i.e., to pick better of the two solutions: one returned by the modified algorithm, and the other the best one-sensor solution [12]. It can be shown the overall algorithm still offers a theoretical performance guarantee for submodular functions, but the performance ratio is reduced by a multiplicative factor of 2. The above model is useful when designing a “load-balanced” strategy to maximize network lifetime of a system—therein, the sensor-selection algorithm must be run periodically based on the remaining battery resources.

**Multiple Transmitters.** Till now, we have implicitly assumed that a single transmitter was present. Our techniques naturally generalize to the case of multiple transmitters, if we represent each *combination* of configurations of present transmitters by a separate hypothesis. Since the MAP, GA, and AGA algorithms are formulated in terms of hypotheses, they generalize naturally to localization of multiple transmitters. However, the key challenge arises due to the large number of hypotheses—exponential in the number of potential transmitters—and thus, the high time complexity of AGA. Fortunately, the techniques discussed in previous subsection can be extended for the case of multiple transmitters as follows.

The key observation is that, for a given hypothesis  $H_i$ , the probability distribution of observations at a sensor  $s$  depends only on the configuration of transmitters in  $H_i$  that within a distance of  $R$  of  $s$ . I.e., for any observation  $x_s$  at a sensor  $s$ ,  $P(x_s|H_{i1}) = P(x_s|H_{i2})$  for any two hypotheses  $H_{i1}$  and  $H_{i2}$  that have the same configuration (locations and powers) for transmitters that are within a distance of  $R$  of  $s$ . The implication of the above observation(s) is that, for a given  $s$ , we can group the given hypotheses into *equivalence classes* based on the configuration of transmitters close of  $s$ , and to compute the benefit of a sensor  $s$  with AGA’s iteration, we only need to compare pairs of equivalence classes (rather than the original hypotheses). The number of such equivalence classes is easily seen to be equal to  $G_R^T$  where  $G_R$  is the number of locations (grid cells) within  $R$  times the number of power levels, and  $T$  is the maximum number of transmitters possible/allowed within a range  $R$  of  $s$  (or any location). Thus, computation of benefit of  $s$  requires consideration of  $G_R^{2T}$  pairs of equivalence classes. If we assume  $T$  to be a small constant, then the overall time complexity of AGA reduces to  $O(nB^3)$  as before, and to  $O(nB)$  if we assume independence of sensor observations.

## IV. EVALUATION

In this section, we evaluate the performance of our algorithms developed in the previous sections. We start with a description of the evaluation platforms used in our experiments.

### A. Implementation

**Implementation Technique.** To evaluate whether AGA runs sufficiently fast to be feasibly used, we implement two distinct versions of AGA using python. The first version, called AGA-Basic, does not utilize the optimizations discussed in Section

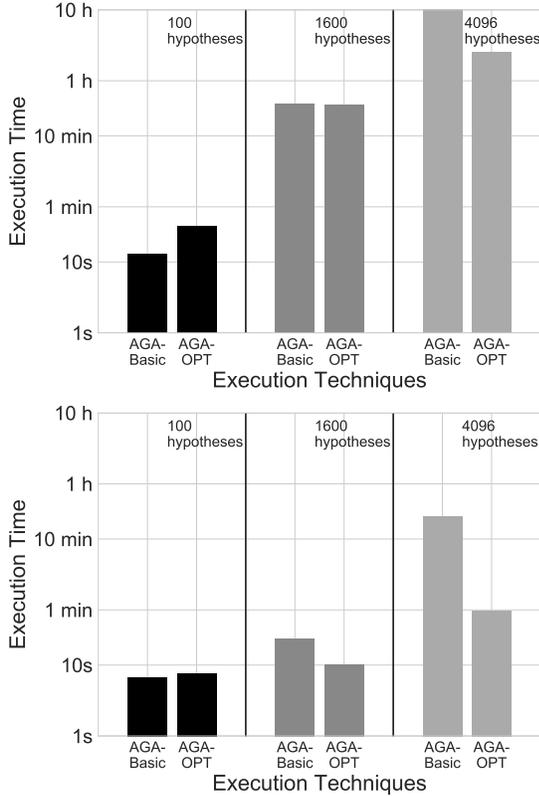


Fig. 4: Execution time of AGA both with and without the optimizations on a (i) CPU and on a (ii) GPU.

III-F. The second version, called AGA-OPT, includes these optimizations. Each version utilizes multiple cores of a CPU using joblib library [13] to compute the gain of each available sensor in parallel. It also uses the numpy library to vectorize operations wherever possible to make execution as fast as possible. We run three different instances of AGA – with 100, 1600 and 4096 hypotheses. Each of these instances have 100 available sensors and a budget of 20. We execute this on a Core i9-7900X CPU having a frequency of 3.30GHz and 20 cores.

**Implementation on CPU.** Figure 4(i) shows the execution time of these three instances. We note that for small instances, the execution time is relatively small. For example, for 100 hypotheses, AGA-basic only takes 13s to execute. However, this rises to 28 minutes for 1600 hypotheses and to over 10 hours for 4096 hypotheses. We also find that for small instances, the optimizations do not lead to much improvement due to the overhead of maintaining the data structures. However, there is a large improvement for 4096 hypotheses, where we get an execution time of 150 minutes using the optimized version.

**Implementation on GPU.** Although execution on CPU’s using our optimizations is feasible, we further note that the bulk of execution time is spent on matrix operations. This suggests that execution on a GPU can lead to much better utilization of data-level parallelism, and further speed up execution. To evaluate this, we optimize the computation of the gain of the sensors using numba library [14] to execute it on a GPU. We utilize an nVidia GTX 1080 GPU having

2560 cores with a processor clock of 1.733 GHz. We then note the execution time for each of the three instances of both AGA-Basic and AGA-OPT.

Figure 4(ii) shows the execution times on a GPU. We note that execution is much faster on a GPU than on a CPU. For example, AGA-OPT now runs in only 8s, 10s and 57s for 100, 1600 and 4096 hypotheses respectively. This shows that AGA can run very fast on a system with GPU, with a speedup of up to 155 times on the large instances.

## B. Evaluation Platforms

We use the following three evaluation platforms with varying fidelity of signal propagation characteristics, to demonstrate the performance of our techniques.

- *Simulation based on synthetic data.* To demonstrate the scalability of our techniques and the sensitivity of our algorithms to changes in settings, we consider a large geographic area of 4km by 4km, with signal path-loss values generated using the SPLAT! application for the Longley-Rice [15]. We use the noise in the sensor measurements (measured independently) to generate the required JPDs. We assume observations to be conditionally independent, thus representing the JPDs as set of probability distributions, one for each sensor and intruder configuration pair. Unless otherwise stated, for this large-scale platform, we use 100m x 100m grid cells giving 1600 potential locations, randomly deploy a transmitter at the height of 30m at a random power between 27-33 dBm which corresponds to roughly 250-750m of transmission range. We randomly deploy 100 spectrum sensors in the area.
- *Indoor Data.* We use publicly available data [16], which deploys transmitters and receivers at 44 locations at an indoor building of an area of 14m x 14m. Here, we use 1m x 1m grid cells, thus giving us a total of 196 potential locations and hypotheses. The transmitters transmit at a frequency of 2.4GHz.
- *Outdoor Testbed.* Finally, to evaluate our techniques in a more practical outdoor setting, we deploy our own testbed in a parking area of dimension 10m x 10m. Each grid cell has size of 1m x 1m. We place a total of 18 sensors on the ground. The sensors consist of single-board computers such as Raspberry Pi’s and Odroid-C2’s connected to an RTL-SDR dongle. The RTL-SDRs use dipole antennas. We collect raw Inphase-Quadrature (I/Q) samples from the RTL-SDRs [17], while transmitting data using a USRP-based transmitter from each grid cell at a height of 1.5m. We perform FFT on the I/Q samples with a bin size of 256 samples to get the signal power values, and then utilize the mean and standard deviation of the power reported for each of the sensors.

**Metrics** We evaluate the performance of a localization strategy in terms of two key metrics: (i) Localization accuracy, i.e.,  $O_{acc}(\mathbf{T})$ , and (ii) Weighted localization error, which weights the misclassification error by the Euclidean distance between the actual and the predicted location (§III-G).

**Compared Algorithms.** We implement both of our designed algorithms, AGA and GA. We also implement two other

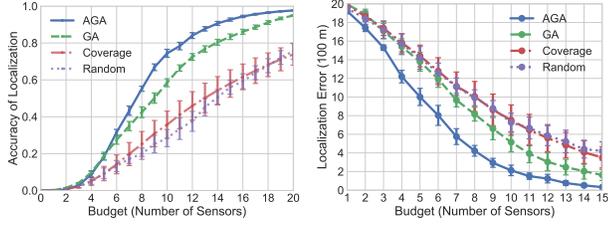


Fig. 5: Comparison of various techniques for (i) Localization accuracy ( $O_{acc}$  ()), and (ii) Weighted localization error, for varying available budget (number of sensors).

techniques for comparison purposes. The first technique, called Coverage, is the selection heuristic from the recent work [2], which essentially tries to maximize the “coverage” of the sensors in a greedy manner.<sup>2</sup> We also implement a Random algorithm which selects the required sensors randomly. We implement these algorithms in python, with extensive use of *numpy* library for vectorized operations. To ensure that our results are statistically significant, we run each of the algorithms 100 times; the range of values is plotted in each of the figures.

### C. Simulation Based on Synthetic Data

**Varying Budget.** Figure 5 shows the performance of our techniques for budgets varying from 1 to 20 sensors. We observe that AGA and GA easily outperform other two algorithms in terms of both metrics, with AGA outperforming even GA quite significantly. For example, AGA outperform Coverage by up to 39% and 56% for localization accuracy and error respectively, while outperforming GA by 15% and 50% for the two metrics respectively.

**Varying Number of Hypotheses.** We now show the performance of our algorithms in terms of localization accuracy, for varying number of hypotheses. In Figure 6, we plot three different cases: (i) the default configuration of 100m by 100m grid cells, (ii) a larger area of 6km by 6km with 100m by 100m grid cells giving 3600 potential locations, and finally (iii) a configuration with default 4km by 4km area, but smaller 62.5m by 62.5m grid cells. First, we observe that AGA continues to outperform other techniques significantly across different cases, with the performance gap between AGA and others (especially GA) increasing with increase in number of hypotheses. Also, as expected, with increase in area and thus number of hypotheses, the accuracy of each of the algorithms falls, but deterioration in AGA’s accuracy is much less compared to other techniques.

**Varying Sensor Density.** Figure 7 shows the accuracy of localization for varying sensor density (i.e., number of available sensors) with a fixed budget of 10 sensors. We note that the accuracy of localization of AGA significantly improves when we increase the number of sensors. For example, it increases by 16% when the number of sensors increases from 50 to 150. In contrast, the performance of GA and Coverage both actually reduces by 7%. This is because having with an increase in

<sup>2</sup>Their approach *Metropolis* performs worse than their greedy approach in open areas [2], and hence, not compared.

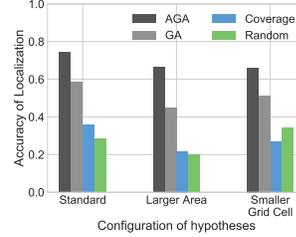


Fig. 6: Comparison for configurations with different number of hypotheses, with a fixed budget of 10 sensors.

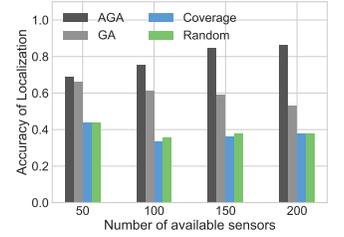


Fig. 7: Comparison for varying number of available sensors, with a fixed budget of 10 sensors.

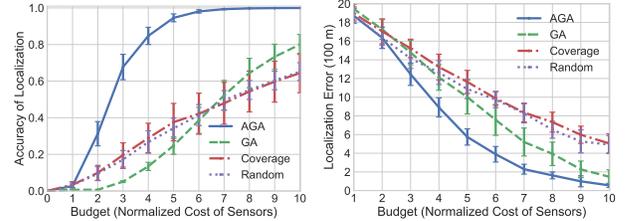


Fig. 8: Comparison of various techniques, for sensors with non-uniform cost.

sensor density GA and Coverage select sensors that are too close to one another to be useful. In contrast, AGA has a submodular objective which leads to an increase in accuracy whenever the value of the optimal value increases.

**Non-Uniform Sensor Costs.** We also evaluate performance of techniques under the setting of sensors with non-uniform cost. We obtain the costs by computing the energy consumption of each sensor by varying the number of samples from 32 to 2048 in multiples of 2. We then randomly assign some energy and corresponding distributions to each sensor. Figure 8 shows the performance for such heterogeneous sensors. As expected, AGA continues to outperform the other techniques in both localization error. However, GA performs much worse than expected in case of heterogeneous sensors.

**Empirical Evaluation of  $k$  Value.** We now evaluate the  $k$  value as defined in Lemma 3. In particular, the performance guarantee of AGA depends on the value of  $k$ , with better performance guarantee for lower  $k$  (ideally,  $k$  should be equal to 1). Figure 9 shows the value of  $k$  for varying budget. We observe that for a very low budget, the value of  $k$  is very large, but it reduces rapidly with increase in budget. In particular, for budgets of 10 and 15 sensors, the value of  $k$  is 1.78 and 1.19 respectively. This shows that AGA’s performance guarantee as per Theorem 3 reaches its near-best for a moderately small budget.

**Comparison with Optimal in Small Instances.** We further confirm AGA’s performance with respect to optimal, we consider small instances of the problem (with 100 hypotheses) and compare AGA with an optimal algorithm. The optimal algorithm uses exhaustive search, which is impossible to execute over larger instances. See Figure 10. We observe that AGA and optimal perform near-identically, with the optimal algorithm yielding at most 0.7% higher localization accuracy than AGA. Note that GA performs worse than AGA and optimal even in this case, albeit by a smaller amount than

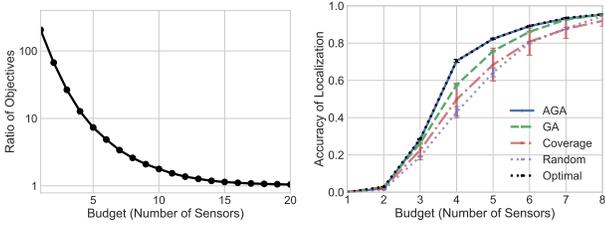


Fig. 9: Values of  $k$  (from Fig. 10: Comparison with an optimal algorithm, for small instances of the problem.

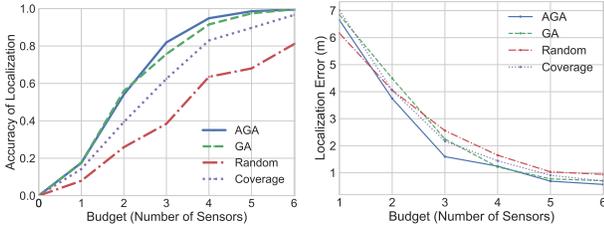


Fig. 11: Performance over public indoor data.

in cases with larger number of hypotheses. Moreover, even for such small instances, the optimal algorithm takes at least an order of magnitude more execution time compared to both AGA and GA.

#### D. Evaluation in Indoor and Outdoor Testbeds

**Indoor Data.** We now evaluate our techniques over a publicly available data-trace taken in an indoor environment, as described in the previous subsection. See Figure 11. We again observe similar performance trends as in previous experiments, for both the performance metrics. The relatively smaller performance gap between AGA and GA is likely due to smaller a number of hypotheses.

**Outdoor Testbed** Figure 12 shows the performance of various algorithms over our outdoor testbed described in the previous subsection. Observe that AGA again performs the best among all techniques in both the metrics. As in the indoor testbed, the performance gap between the AGA and GA is less compared to the large-scale simulations due to a small number of hypotheses. Note that because of the noise in the dataset, the localization accuracy reaches a maximum of only 75% even with all the 18 sensors.

## V. RELATED WORK

**Sensor Selection for Transmitter Localization.** A large number of works have developed techniques for detecting and localizing transmitters or intruders that emit radio signals [18], [19]. Note that the transmitter localization problem is slightly different from another well-studied problem of indoor localization [20], wherein a user is localized based on signal *received* from multiple transmitters; herein, the issue of selecting transmitters to localize the user has no incentive, and hence not been addressed before. To the best of our knowledge, none of these prior works on transmitter localization either have addressed the optimization problem addressed in the paper. The closest related works are [1] and [2] as discussed

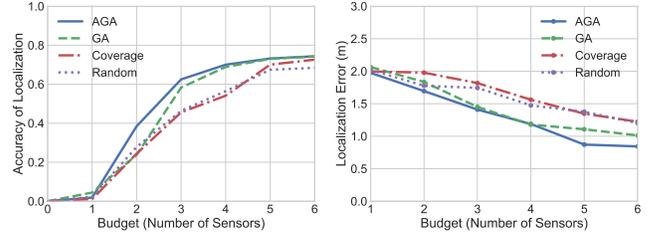


Fig. 12: Performance over outdoor testbed data.

next. The work [1] focuses on *detection* of unauthorized transmitters using low-cost sensors in the context of shared spectrum systems; they consider the problem of selection of sensors in this context, and propose a heuristic with no performance guarantees. The key difference of our work from theirs is that they focus on detection of transmitters, which is a much simpler problem than localization of transmitters. In addition, [2] considers selection of sensors for transmitter localization, but with a objective of maximizing the “coverage” of the region by the sensors. They present heuristics without any performance guarantees. Nevertheless, we implement their approach and compare with our techniques (§IV).

**Sensor Selection in Sensor Networks.** Sensor selection is a natural problem to address in the context of wireless sensor networks deployed to detect and/or localize an event or phenomenon (see [21] for a survey). Many of these works have leverage the submodularity property to develop greedy approximation algorithms. The closest work among these is that of [22] which shows approximability of the greedy approach for the problem of minimizing uncertainty in estimating a spatial phenomenon (e.g., temperature). However, in general, the key difference of our work with these works is our desired objective function ( $O_{\text{acc}}$  or  $P_{\text{err}}$ )—and thus, the making the proof of monotonicity and/or submodularity of the objective function very different. In our case, we had to even circumvent the non-submodularity of the objective function  $O_{\text{acc}}$  by considering an appropriate auxiliary objective function.

**Online Selection of Sensors.** An alternate formulation of our sensor selection problem could be to select sensors adaptively *based* on the observations of previously selected sensors. This online problem is similar to the adaptive stochastic optimization problem addressed in other contexts [23], [24], [25], [26]. However, in online selection, a sensor is selected based on analysis (which will incur non-trivial latency) of observations of previous sensors. This makes localization based on *near-simultaneous* sensor observations, required to localize intermittent transmitters, infeasible. Also, note that online selection needs to be done anew for each localization, which may be performed very frequently (e.g., every second or fraction of a second) in many applications, e.g., spectrum patrolling. Thus, our focus is on offline selection.

## VI. CONCLUSION

In this work, we have considered the hypothesis-driven approach for localization of transmitters, and developed techniques to optimize the localization accuracy under a constraint of limited resources. Developed techniques have been shown

to yield provably approximate solutions. Our work can be instrumental in maximizing the network lifetime of a spectrum monitoring and/or patrolling system. Our future work focusses on improving our theoretical performance guarantee results, and developing similar sensor selection approximation algorithms for other localization approaches that are not hypothesis-driven.

#### ACKNOWLEDGMENTS

This work is supported by the NSF grant CNS-1642965.

#### REFERENCES

- [1] A. Bhattacharya, A. Chakraborty, S. R. Das, H. Gupta, and P. M. Djurić, "Spectrum patrolling with crowdsourced spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, (accepted) 2019. doi: 10.1109/TCCN.2019.2939793
- [2] M. Khaledi, M. Khaledi, S. Sarkar, S. Kasera, N. Patwari, K. Derr, and S. Ramirez, "Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking - MobiCom*, 2017. doi: 10.1145/3117811.3117845 p. 235–247.
- [3] A. Nika, Z. Li, Y. Zhu, Y. Zhao, B. Y. Zhao, X. Zhou, and H. Zheng, "Empirical validation of commodity spectrum monitoring," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems - SenSys*, 2016. doi: 10.1145/2994551.2994557 p. 96–108.
- [4] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das, "SpecSense: Crowdsensing for efficient querying of spectrum occupancy," in *IEEE INFOCOM - IEEE Conference on Computer Communications*, may 2017. doi: 10.1109/infocom.2017.8057113 pp. 1–9.
- [5] V. Cevher, P. Boufounos, R. G. Baraniuk, A. C. Gilbert, and M. J. Strauss, "Near-optimal bayesian localization via incoherence and sparsity," in *International Conference on Information Processing in Sensor Networks - IPSN*, Apr 2009, pp. 205–216.
- [6] A. Bhattacharya, C. Zhan, , H. Gupta, S. R. Das, and P. M. Djurić, "Selection of sensors for efficient transmitter localization," in *IEEE INFOCOM*, 2020.
- [7] C. Zhan, H. Gupta, A. Bhattacharya, and M. Ghaderibaneh, "Efficient localization of multiple intruders for shared spectrum system," in *International Conference on Information Processing in Sensor Networks - IPSN*, Apr 2020.
- [8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.
- [9] P. K. Penumarthi, A. Q. Li, J. Banfi, N. Basilico, F. Amigoni, J. O’Kane, I. Rekleitis, and S. Nelakuditi, "Multirobot exploration for building communication maps with prior from communication models," in *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, Dec 2017. doi: 10.1109/mrs.2017.8250936 pp. 90–96.
- [10] "NSF workshop on spectrum measurements infrastructure," p. 22–23, 2016.
- [11] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, 1978. doi: 10.1007/BF01588971
- [12] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39 – 45, 1999. doi: 10.1016/S0020-0190(99)00031-9
- [13] Joblib, "joblib/joblib," Oct 2018. [Online]. Available: <https://github.com/joblib/joblib>
- [14] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-based Python JIT Compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 2015. doi: 10.1145/2833157.2833162 pp. 1 – 6.
- [15] K. Chamberlin and R. Luebbers, "An evaluation of Longley-Rice and GTD propagation models," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 6, pp. 1093 – 1098, 1982. doi: 10.1109/TAP.1982.1142958
- [16] N. Patwari, "CRAWDAD dataset utah/cir (v. 2007-09-10)," Downloaded from <https://crawdad.org/utah/CIR/20070910>, Sep. 2007.
- [17] "RTL-SDR (RTL2832U) and software defined radio news and projects," <https://www.rtl-sdr.com/>, 2018, accessed on Oct 18, 2018.
- [18] R. K. Martin and R. Thomas, "Algorithms and bounds for estimating location, directionality, and environmental parameters of primary spectrum users," *IEEE Transactions on Wireless Communications*, vol. 8, no. 11, pp. 5692–5701, 2009. doi: 10.1109/TWC.2009.090494
- [19] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, thirdquarter 2019. doi: 10.1109/COMST.2019.2911558
- [20] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2017. doi: 10.1109/comst.2016.2632427
- [21] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-Noy, T. Brown, and T. La Porta, "A survey of sensor selection schemes in wireless sensor networks," in *Unattended Ground, Sea, and Air Sensor Technologies and Applications IX*, vol. 6562. International Society for Optics and Photonics, 2007.
- [22] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, 2008.
- [23] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *International Conference on Information Processing in Sensor Networks - IPSN*, 2004. doi: 10.1145/984622.984628 pp. 36–45.
- [24] Y. Chen, S. H. Hassani, A. Karbasi, and A. Krause, "Sequential information maximization: When is greedy near-optimal?" in *Proceedings of The 28th Conference on Learning Theory*, vol. 40, 03–06 Jul 2015, pp. 338–363. [Online]. Available: <http://proceedings.mlr.press/v40/Chen15b.html>
- [25] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, ser. UAI’05, Arlington, Virginia, USA, 2005, p. 324–331.
- [26] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *Journal of Artificial Intelligence Research*, vol. 42, no. 1, p. 427–486, Sep. 2011.
- [27] E. Seneta, "On the history of the strong law of large numbers and boole’s inequality," *Historia Mathematica*, vol. 19, no. 1, pp. 24 – 39, 1992. doi: 10.1016/0315-0860(92)90053-E
- [28] M. Abramowitz, I. A. Stegun *et al.*, "Handbook of mathematical functions with formulas, graphs, and mathematical tables," 1972.

# Appendix

## APPENDIX A PROOF OF NP-HARDNESS

**Lemma.** *The Offline Sensor Selection (LSS) problem is NP-Hard.*

*Proof.* In the maximum coverage problem, we are given a universe set  $\mathbf{U} = \{u_k\}$ , and a collection of its subsets,  $\mathbf{V}_k \subseteq \mathbf{U}$ . The objective is to choose at most a budget  $B$  number of  $\mathbf{V}_k$ 's so that the number of items in their union is maximized. To map this to LSS, we set  $\mathbf{U}$  equal to the set of hypotheses  $H_0$  to  $H_{m-1}$ . Also, we have a different  $\mathbf{V}_k$  for each  $s_k \in \mathbf{S}$ . Let the corresponding sensor observation be denoted by  $\mathbf{x}_k$ . For each item  $u_i \in \mathbf{V}_k$ , we set the value of  $P(\mathbf{x}_k|H_i) = 1$ . For each item  $u_i \in \mathbf{U} \setminus \mathbf{V}_k$ , we get  $P(\mathbf{x}_k|H_i) = 0$ . Now, let  $\mathbf{V}_k = \{H_i : P(\mathbf{x}_k|H_i) = 1\}$ . Thus, we have an LSS problem instance, with a given JPD for all sensors, fixed number of hypotheses and a fixed budget  $B$ . Solving this instance returns a fixed subset of sensors  $\mathbf{T}$ . Since a single sensor corresponds to a single subset  $\mathbf{V}_k$ , we select the subsets corresponding to the selected sensors. Thus, we can solve an arbitrary maximum coverage problem if LSS is solvable in polynomial time. This shows that the LSS problem is NP-Hard.  $\square$

## APPENDIX B PROOF OF THEOREM 1

**Theorem 1.** *Given the hypotheses  $H_i \sim N(\mathbf{p}_i, \Sigma)$  and  $H_j \sim N(\mathbf{p}_j, \Sigma)$ , a data vector  $\mathbf{x} = [x_1, \dots, x_{|\mathbf{T}|}]^T$  can be classified by applying the following threshold test:*

$$\mathbf{x}^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) \underset{H_i}{\overset{H_j}{\gtrless}} \frac{1}{2} (\mathbf{p}_j + \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) + \log \frac{P(H_i)}{P(H_j)} \quad (1)$$

*Proof.* We have two hypotheses  $H_i$  and  $H_j$ . We first compute the posterior probabilities as follows:

$$\begin{aligned} \frac{P(H_j|\mathbf{x})}{P(H_i|\mathbf{x})} &= \frac{P(\mathbf{x}|H_j)P(H_j)}{P(\mathbf{x}|H_i)P(H_i)} \quad [\text{Using Bayes' Theorem}] \\ &= \frac{P(x_1, \dots, x_{|\mathbf{T}|}|H_j)P(H_j)}{P(x_1, \dots, x_{|\mathbf{T}|}|H_i)P(H_i)} \\ &= \frac{P(H_j)}{P(H_i)} \frac{\exp[-\frac{1}{2}(\mathbf{x} - \mathbf{p}_j)^T \Sigma^{-1} (\mathbf{x} - \mathbf{p}_j)]}{\exp[-\frac{1}{2}(\mathbf{x} - \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{x} - \mathbf{p}_i)]} \\ &= \frac{P(H_j)}{P(H_i)} \exp[-\frac{1}{2}(\mathbf{x} - \mathbf{p}_j)^T \Sigma^{-1} (\mathbf{x} - \mathbf{p}_j) \\ &\quad + \frac{1}{2}(\mathbf{x} - \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{x} - \mathbf{p}_i)] \end{aligned} \quad (2)$$

We now consider the ratio test:

$$\frac{P(H_j|\mathbf{x})}{P(H_i|\mathbf{x})} \underset{H_i}{\overset{H_j}{\gtrless}} 1 \quad (3)$$

Taking logarithm on both sides of Eqn. (2), we get:

$$-\frac{1}{2}(\mathbf{x} - \mathbf{p}_j)^T \Sigma^{-1} (\mathbf{x} - \mathbf{p}_j) + \frac{1}{2}(\mathbf{x} - \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{x} - \mathbf{p}_i) \underset{H_i}{\overset{H_j}{\gtrless}} \log \frac{P(H_i)}{P(H_j)} \quad (4)$$

which on simplifying gives us:

$$\begin{aligned} (\mathbf{x} - \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{x} - \mathbf{p}_i) - (\mathbf{x} - \mathbf{p}_j)^T \Sigma^{-1} (\mathbf{x} - \mathbf{p}_j) &\underset{H_i}{\overset{H_j}{\gtrless}} 2 \log \frac{P(H_i)}{P(H_j)} \\ \implies \mathbf{x}^T \Sigma^{-1} \mathbf{p}_j + \mathbf{p}_j^T \Sigma^{-1} \mathbf{x} - \mathbf{p}_j^T \Sigma^{-1} \mathbf{p}_j - \mathbf{x}^T \Sigma^{-1} \mathbf{p}_i \\ &\quad - \mathbf{p}_i^T \Sigma^{-1} \mathbf{x} + \mathbf{p}_i^T \Sigma^{-1} \mathbf{p}_i \underset{H_i}{\overset{H_j}{\gtrless}} 2 \log \frac{P(H_i)}{P(H_j)} \\ \implies 2\mathbf{x}^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) + \mathbf{p}_i^T \Sigma^{-1} \mathbf{p}_i - \mathbf{p}_j^T \Sigma^{-1} \mathbf{p}_j &\underset{H_i}{\overset{H_j}{\gtrless}} 2 \log \frac{P(H_i)}{P(H_j)} \\ \implies \mathbf{x}^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) \underset{H_i}{\overset{H_j}{\gtrless}} \frac{1}{2} (\mathbf{p}_j + \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) + \log \frac{P(H_i)}{P(H_j)} \end{aligned} \quad (5)$$

This proves our theorem. In the special case of equal priors, i.e.,  $P(H_i) = P(H_j)$ , Eqn. (5) further simplifies to

$$\mathbf{x}^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) \underset{H_i}{\overset{H_j}{\gtrless}} \frac{1}{2} (\mathbf{p}_j + \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) \quad (6)$$

For convenience, we henceforth denote  $\mathbf{p} = \mathbf{p}_j - \mathbf{p}_i$ .  $\square$

We call the LHS of Eqn. (1) as test statistic  $T(\mathbf{x})$ .

**Corollary 1.** *If  $H_i$  is true, then  $T(\mathbf{x})$  follows the Gaussian distribution  $N(\mathbf{p}_i^T \Sigma^{-1} \mathbf{p}, \mathbf{p}^T \Sigma^{-1} \mathbf{p})$ . If  $H_j$  is true, then  $T(\mathbf{x})$  follows the Gaussian distribution  $N(\mathbf{p}_j^T \Sigma^{-1} \mathbf{p}, \mathbf{p}^T \Sigma^{-1} \mathbf{p})$*

*Proof.* We first find the mean of  $T(\mathbf{x})$ . We note that:

$$\begin{aligned} E[T(\mathbf{x})] &= E[\mathbf{x}^T \Sigma^{-1} \mathbf{p}] = E[\mathbf{x}]^T \Sigma^{-1} \mathbf{p} \\ &= \begin{cases} \mathbf{p}_i^T \Sigma^{-1} \mathbf{p}, & \text{if } H_i \text{ is true} \\ \mathbf{p}_j^T \Sigma^{-1} \mathbf{p}, & \text{if } H_j \text{ is true} \end{cases} \end{aligned} \quad (7)$$

To find the variance of  $T(\mathbf{x})$ , we denote the mean of  $T(\mathbf{x})$  to be  $p_k$ , where  $k \in i, j$  depending on whether hypotheses  $H_i$  or  $H_j$  is true. We also note that the covariance matrix  $\Sigma$  is symmetric positive definite, i.e.,  $\Sigma = \Sigma^T$ .

$$\begin{aligned} \text{Var}[T(\mathbf{x})] &= E[T(\mathbf{x}) - E[T(\mathbf{x})]]^2 \\ &= E[\mathbf{x}^T \Sigma^{-1} \mathbf{p} - \mathbf{p}_k^T \Sigma^{-1} \mathbf{p}]^2 \\ &= E[(\mathbf{x} - \mathbf{p}_k)^T \Sigma^{-1} \mathbf{p}]^2 \\ &= E[\mathbf{p} \Sigma^{-1} (\mathbf{x} - \mathbf{p}_k) (\mathbf{x} - \mathbf{p}_k)^T \Sigma^{-1} \mathbf{p}] \\ &\quad [\text{as } (\Sigma^{-1})^T = (\Sigma^T)^{-1} = \Sigma^{-1}] \\ &= E[\mathbf{p}^T \Sigma^{-1} \Sigma \Sigma^{-1} \mathbf{p}] \\ &= \mathbf{p}^T \Sigma^{-1} \mathbf{p} \end{aligned} \quad (8)$$

Thus, the statement of Corollary 1 follows.  $\square$

APPENDIX C  
PROOFS OF LEMMAS 1 AND 2

We first derive an expression of  $O_{\text{acc}}(\mathbf{T})$ . We note that:

$$\begin{aligned}
P(\text{MAP}(\mathbf{x}) = i \mid H_i) &= P\left(T(\mathbf{x}) \leq \frac{1}{2}(\mathbf{p}_j + \mathbf{p}_i)^T \Sigma^{-1}(\mathbf{p}_j - \mathbf{p}_i) \mid T(\mathbf{x})\right) \\
&\quad \sim N(\mathbf{p}_i^T \Sigma^{-1} \mathbf{p}, \mathbf{p}^T \Sigma^{-1} \mathbf{p}) \\
&= P\left(\frac{T(\mathbf{x}) - \mathbf{p}_i^T \Sigma^{-1} \mathbf{p}}{\sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}} \leq \frac{1}{2} \sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}\right) \\
&\quad \left| \frac{T(\mathbf{x}) - \mathbf{p}_i^T \Sigma^{-1} \mathbf{p}}{\sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}} \sim N(0, 1) \right) \\
&= P\left(Z \leq \frac{1}{2} \sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}} \mid Z \sim N(0, 1)\right) \\
&\quad \left[ \text{introducing new random variable, } Z = \frac{T(\mathbf{x}) - \mathbf{p}_i^T \Sigma^{-1} \mathbf{p}}{\sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}} \right] \\
&= \Phi\left(\frac{1}{2} \sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}\right) \tag{9}
\end{aligned}$$

where  $\Phi$  is the CDF of standard normal distribution. Also,

$$\begin{aligned}
P(\text{MAP}(\mathbf{x}) = j \mid H_j) &= P\left(T(\mathbf{x}) \geq \frac{1}{2}(\mathbf{p}_j + \mathbf{p}_i)^T \Sigma^{-1}(\mathbf{p}_j - \mathbf{p}_i) \mid T(\mathbf{x})\right) \\
&\quad \sim N(\mathbf{p}_j^T \Sigma^{-1} \mathbf{p}, \mathbf{p}^T \Sigma^{-1} \mathbf{p}) \\
&= P\left(\frac{T(\mathbf{x}) - \mathbf{p}_j^T \Sigma^{-1} \mathbf{p}}{\sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}} \geq -\frac{1}{2} \sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}\right) \\
&\quad \left| \frac{T(\mathbf{x}) - \mathbf{p}_j^T \Sigma^{-1} \mathbf{p}}{\sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}} \sim N(0, 1) \right) \\
&= P\left(Z \leq \frac{1}{2} \sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}} \mid Z \sim N(0, 1)\right) \\
&\quad \left[ \text{introducing new random variable, } Z = \frac{T(\mathbf{x}) - \mathbf{p}_j^T \Sigma^{-1} \mathbf{p}}{\sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}} \right] \\
&= \Phi\left(\frac{1}{2} \sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}\right) \tag{10}
\end{aligned}$$

This gives us the following expression of  $O_{\text{acc}}(\mathbf{T})$ :

$$\begin{aligned}
O_{\text{acc}}(\mathbf{T}) &= P(\text{MAP}(\mathbf{x}) = i \mid H_i)P(H_i) \\
&\quad + P(\text{MAP}(\mathbf{x}) = j \mid H_j)P(H_j) \\
&= \Phi\left(\frac{1}{2} \sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}\right)[P(H_i) + P(H_j)] \\
&= \Phi\left(\frac{1}{2} \sqrt{\mathbf{p}^T \Sigma^{-1} \mathbf{p}}\right) \quad [\text{as } P(H_i) + P(H_j) = 1] \tag{11}
\end{aligned}$$

We use this expression of  $O_{\text{acc}}(\mathbf{T})$  to get the following lemmas:

**Lemma 1.** *The objective  $O_{\text{acc}}(\mathbf{T})$  is monotone in nature, i.e. if some sensor  $s_k \in \mathbf{S} \setminus \mathbf{T}$ , then  $O_{\text{acc}}(\mathbf{T} \cup \{s_k\}) \geq O_{\text{acc}}(\mathbf{T})$ .*

*Proof.* We show that the argument of  $\Phi$  increases when we add a sensor  $s_k$  to  $\mathbf{T}$ . To do this, we first denote the covariance matrices of  $\mathbf{T}$  and  $\mathbf{T} \cup \{s_k\}$  by  $\Sigma$  and  $\Sigma_{\text{new}}$  respectively. Also, let:

$$\Sigma_{\text{new}} = \begin{bmatrix} \Sigma & b_k \\ b_k^T & \sigma_k^2 \end{bmatrix} \tag{12}$$

where the vector  $b_k$  represents the noise correlation between the sensors  $s_k$  and  $s_i \forall i \in [1, \dots, |\mathbf{T}|]$  and  $\sigma_k^2$  is the noise covariance of the sensor  $s_k$ . Using Banachiewicz inversion [1], we get:

$$\begin{aligned}
\Sigma_{\text{new}}^{-1} &= \begin{bmatrix} \Sigma^{-1} & O \\ O & O \end{bmatrix} \\
&\quad + \begin{bmatrix} -\Sigma^{-1} b_k \\ I \end{bmatrix} (\sigma_k^2 - b_k^T \Sigma^{-1} b_k)^{-1} \begin{bmatrix} -\Sigma^{-1} b_k & I \end{bmatrix} \tag{13}
\end{aligned}$$

Since  $\Sigma_{\text{new}} \succeq 0$  and  $\Sigma \succeq 0$ , the Schur complement of  $\Sigma$  in  $\Sigma_{\text{new}}$  is also positive semidefinite, i.e.,

$$\Sigma_{\text{new}} / \Sigma = (\sigma_k^2 - b_k^T \Sigma^{-1} b_k) \geq 0. \tag{14}$$

With the addition of sensor  $s_k$  to the subset  $\mathbf{T}$ , the mean vector can be rewritten as  $p_{\text{new}} = [p^T \quad \mu_k]^T$ . Now the argument of  $\Phi$  can be computed as:

$$\begin{aligned}
p_{\text{new}}^T \Sigma_{\text{new}}^{-1} p_{\text{new}} &= [p^T \quad \mu_k] \begin{bmatrix} \Sigma^{-1} & O \\ O & O \end{bmatrix} \begin{bmatrix} p \\ \mu_k \end{bmatrix} + a_k^2 (\sigma_k^2 - b_k^T \Sigma^{-1} b_k)^{-1} \\
&= p^T \Sigma^{-1} p + a_k^2 (\sigma_k^2 - b_k^T \Sigma^{-1} b_k)^{-1} \tag{15}
\end{aligned}$$

where the scaler  $a_k$  is given by

$$a_k = [p^T \quad \mu_k] \begin{bmatrix} -\Sigma^{-1} b_k \\ I \end{bmatrix} = [-\Sigma^{-1} b_k \quad I] \begin{bmatrix} p \\ \mu_k \end{bmatrix} \tag{16}$$

From Eqn. (14) and Eqn. (15), we have

$$p_{\text{new}}^T \Sigma_{\text{new}}^{-1} p_{\text{new}} \geq p^T \Sigma^{-1} p \tag{17}$$

Thus, adding a dimension to  $\mathbf{p}$  can never reduce the value of  $p^T \Sigma^{-1} p$ . Since the CDF of the standard normal distribution  $\Phi$  is a non-decreasing function in its argument,  $O_{\text{acc}}(\mathbf{T})$  increases by adding sensors to  $\mathbf{T}$ .  $\square$

**Lemma 2.** *The objective  $O_{\text{acc}}(\mathbf{T})$  is submodular in nature, i.e. if some sensor  $s_k \in \mathbf{S} \setminus \mathbf{T}_2$ , where  $\forall \mathbf{T}_1 \subseteq \mathbf{T}_2 \subseteq \mathbf{S}$ , we have:*

$$O_{\text{acc}}(\mathbf{T}_1 \cup \{s_k\}) - O_{\text{acc}}(\mathbf{T}_1) \geq O_{\text{acc}}(\mathbf{T}_2 \cup \{s_k\}) - O_{\text{acc}}(\mathbf{T}_2) \tag{18}$$

*Proof.* We denote the difference of the values of  $O_{\text{acc}}$  as gain  $G$ , and define it as follows:

$$G(\mathbf{T}, s_k) = O_{\text{acc}}(\mathbf{T} \cup \{s_k\}) - O_{\text{acc}}(\mathbf{T}) \tag{19}$$

Now, let us consider two subsets of  $\mathbf{S}$ ,  $\mathbf{T}_1$  and  $\mathbf{T}_2$ , where  $\mathbf{T}_1 \subseteq \mathbf{T}_2$ . Note that in this case,  $O_{\text{acc}}(\mathbf{T}_1) \leq O_{\text{acc}}(\mathbf{T}_2)$  due to monotonicity. We also note that the double derivative of  $O_{\text{acc}}$  is negative, and that the  $O_{\text{acc}}$  function is continuous and differentiable at all points. Thus, from the nature of  $\phi$  function, and the fact that  $O_{\text{acc}}$  is monotone, we observe that the gain  $G(\mathbf{T}_1, s_k) \geq G(\mathbf{T}_2, s_k)$ . This proves our theorem.  $\square$

APPENDIX D  
COUNTER-EXAMPLE TO SHOW THAT ACTUAL OBJECTIVE  
IS NOT SUBMODULAR

We consider the case where there are multiple hypotheses, where each hypothesis  $H_i (i = 0, \dots, m)$  has a multi-variate Gaussian distribution with its individual mean but the same covariance. Formally,

$$H_i : x \sim N(\mathbf{p}_i, \Sigma) \quad (20)$$

For convenience, we call this setting as **Gaussian input with multiple hypotheses**. We now show that in this setting, the objective  $O_{\text{acc}}$  is not submodular.

**Lemma.** *In a setting with multiple hypotheses,  $O_{\text{acc}}$  is not submodular.*

*Proof.* We have the following expression of  $O_{\text{acc}}$ :

$$O_{\text{acc}} = \sum_{i=0}^m \prod_{j \neq i} [1 - P(\text{MAP}_{ij}(\mathbf{x}) \neq j | H_i)] P(H_i) \quad (21)$$

Note that since  $\text{MAP}_{ij}$  is the test among the two hypotheses:

$$H_i : N(\mathbf{p}_i, \Sigma) \quad \& \quad H_j : N(\mathbf{p}_j, \Sigma) \quad (22)$$

We first shift the means of both hypotheses so that the mean of  $H_i$  is set to  $O$ . This does not affect the probability of misclassification, since both the means are equally shifted. Then,  $H_i$  and  $H_j$  have means of  $O$  and  $\mathbf{p}_j - \mathbf{p}_i$  respectively. Now, using Lemma 2, we have the value of  $O_{\text{acc}}$  as:

$$O_{\text{acc}} = \sum_{i=0}^m P(H_i) \left[ \prod_{j \neq i} \left( 1 - Q\left(\frac{1}{2} \sqrt{(\mathbf{p}_j - \mathbf{p}_i)^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i)}\right) \right) \right] \quad (23)$$

We now show that  $O_{\text{acc}}$  is not submodular using a counter-example. Let there be three hypothesis  $H_0, H_1$  and  $H_2$  with prior probabilities  $P(H_i)$  each equal to 0.33 and two sensors with the mean vectors  $[0, 0], [0.75, 0.75]$  and  $[0.5, 0.5]$ . Also assume that  $\Sigma$  is an identity matrix. A realistic scenario where this configuration is possible is shown in Figure 1.

We first observe that when no sensors are selected, we select one among the three hypothesis at random, which will be correct only with an expected probability of 0.33, i.e.,  $O_{\text{acc}}(\{\}) = 0.33$ . We now show the values of  $O_{\text{acc}}$ , which is also shown visually in Figure 2.

$$O_{\text{acc}}(\{s_1\}) = 0.3571 \quad (24)$$

Thus, the gain  $G(s_1, \{\}) = O_{\text{acc}}(s_1) - O_{\text{acc}}(\{\}) = 0.023$ . Now, we compute the gain of adding the second sensor. Selecting both sensors, we get the value of  $O_{\text{acc}}$  as:

$$O_{\text{acc}}(\{s_1, s_2\}) = 0.4041 \quad (25)$$

Thus, the gain  $G(s_2, \{s_1\}) = O_{\text{acc}}(\{s_1, s_2\}) - O_{\text{acc}}(\{s_1\}) = 0.0469$ . We observe that the gain has gone up from 0.023 to 0.0469 on adding the sensor  $s_2$  to our set  $\{s_1, s_2\}$ . Thus, the objective  $O_{\text{acc}}$  is not submodular.  $\square$

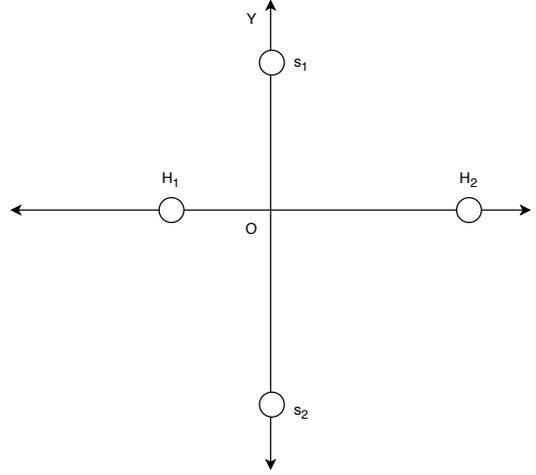


Fig. 1: A schematic representation of the configuration of sensors and hypotheses described in Appendix D. The location corresponding to hypothesis  $H_1$  is closer to the origin as compared to  $H_2$ , since higher power is sensed if it is true. Note that  $H_0$  is not shown, as it denotes absence of any transmitter.

APPENDIX E  
PROOF OF LEMMA 3

First, we recall from Section 3 that:

$$O_{\text{acc}}(\mathbf{T}) = \sum_{i=0}^{m-1} [1 - P(\bigcup_{j \neq i} \text{MAP}_{ij}(\mathbf{x}) = j | H_i)] P(H_i), \quad \text{and} \quad (26)$$

$$O_{\text{aux}}(\mathbf{T}) = 1 - \sum_{i=0}^{m-1} \sum_{j \neq i} P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i) P(H_i) \quad (27)$$

We prove the lemma in three parts.

$O_{\text{aux}}(\mathbf{T}) \leq O_{\text{acc}}(\mathbf{T})$ . This directly follows from an application of Boole's inequality [2] which states that the probability of a union of events is never greater than the sum of the probabilities of individual events. In particular, by Boole's inequality, we have for all  $i$ :

$$P(\bigcup_{j \neq i} \text{MAP}_{ij} = j | H_i) \leq \sum_{j \neq i} P(\text{MAP}_{ij} = j | H_i) \quad (28)$$

Then, by multiplying each by  $P(H_i)$ , summing over all  $i$ , subtracting each side from 1, and noting that  $\sum_i P(H_i) = 1$ , we get  $O_{\text{aux}}(\mathbf{T}) \leq O_{\text{acc}}(\mathbf{T})$  using Eq (26) and Eq (27).

$O_{\text{acc}}(\mathbf{T}) \leq 1 - \frac{1}{k}(1 - O_{\text{aux}}(\mathbf{T}))$ . To get this, we utilize the fact that the probability of a union of events is more than the probability of each of the individual events. Thus,

$$P(\bigcup_{j \neq i} \text{MAP}_{ij}(\mathbf{x}) = j | H_i) \geq \max_{j \neq i} \{P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i)\} \quad \forall i.$$

We also have the below, as maximum is greater than mean:

$$\max_{j \neq i} \{P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i)\} \geq \frac{1}{m} \sum_{j \neq i} P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i) \quad \forall i,$$

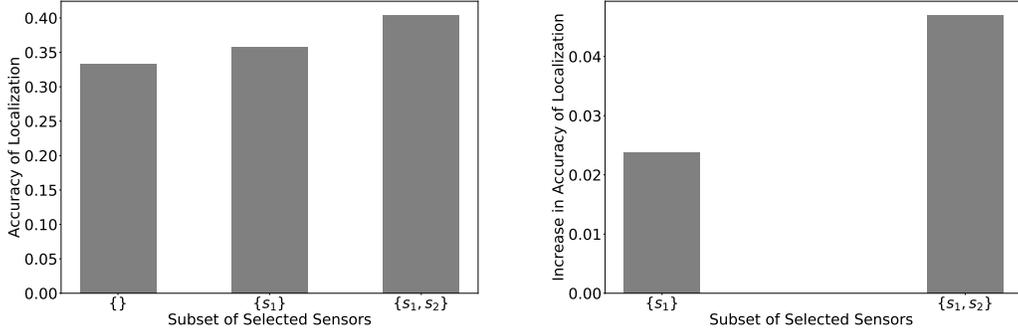


Fig. 2: In the configuration shown in Appendix D, we show (a) value of  $O_{acc}$  for different subsets, and (b) increase in value of  $O_{acc}$ .

where  $0 \leq i \leq m$ . Now, using Eq (26) and the above two equations, we get:

$$\begin{aligned} O_{acc}(\mathbf{T}) &\leq 1 - \frac{1}{m} \sum_{i=0}^m \sum_{j \neq i} P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i) P(H_i) \\ &= 1 - \frac{1}{m} (1 - O_{aux}(\mathbf{T})). \end{aligned}$$

**Value of  $k$  reduces with increase in  $O_{acc}$ .** We need to show that:

$$\frac{1 - O_{aux}(T')}{1 - O_{acc}(T')} \leq \frac{1 - O_{aux}(T)}{1 - O_{acc}(T)}, \text{ where } T' \supseteq T \quad (29)$$

We show the more general case. For any events,  $A, B$ , if their individual probabilities reduce the ratio of their sum and union also reduces, i.e.

$$\frac{P(A_1) + P(B_1)}{P(A_1 \cup B_1)} \geq \frac{P(A_2) + P(B_2)}{P(A_2 \cup B_2)}, \quad (30)$$

where  $P(A_1) \geq P(A_2)$  &  $P(B_1) \geq P(B_2)$

Let  $\Delta_A = P(A_1) - P(A_2)$  and  $\Delta_B = P(B_1) - P(B_2)$ . Then, we have:

$$\begin{aligned} &\frac{P(A_2) + P(B_2)}{P(A_2 \cup B_2)} \\ &= \frac{P(A_1) - \Delta_A + P(B_1) - \Delta_B}{P(A_1) - \Delta_A + P(B_1) - \Delta_B - (P(A_1) - \Delta_A)(P(A_2) - \Delta_B)} \\ &= \frac{P(A_1) + P(B_1) - \Delta}{P(A_1) + P(B_1) - \Delta - (P(A_1) - \Delta_A)(P(B_1) - \Delta_B)} \\ &= \frac{P(A_1) + P(B_1) - \Delta}{P(A_1 \cup B_1) - \Delta + (P(B_1)\Delta_A + P(A_1)\Delta_B - \Delta_A\Delta_B)} \\ &\leq \frac{P(A_1) + P(B_1)}{P(A_1 \cup B_1)} \end{aligned}$$

Note that the last term in the denominator is a positive term. This shows that the denominator has increased more, and the value of the overall fraction has reduced. We can easily generalize this to more than two events. We note that both  $O_{acc}(T') \geq O_{acc}(T)$ , and  $O_{aux}(T') \geq O_{aux}(T)$ . Thus, Eqn. (29) follows.

## APPENDIX F PROOF OF THEOREM 3

Let  $\mathbf{T}$  be AGA solution, and  $\mathbf{T}'$  be any solution. We have:

$$\begin{aligned} O_{aux}(\mathbf{T}) &\geq 0.63O_{aux}(\mathbf{T}') \\ (1 - O_{aux}(\mathbf{T})) &\leq 0.63(1 - O_{aux}(\mathbf{T}')) + 0.37 \\ (1 - O_{acc}(\mathbf{T})) &\leq 0.63k(1 - O_{acc}(\mathbf{T}')) + 0.37 \\ P_{err}(\mathbf{T}) &\leq 0.63kP_{err}(\mathbf{T}') + 0.37 \end{aligned}$$

We have used Lemma 3 in the third equation above. Let  $\mathbf{T}'$  be the solution with optimal  $O_{acc}()$  (and thus, optimal  $P_{err}$ ), and the lemma follows.

## APPENDIX G INDEPENDENT SENSOR OBSERVATIONS

From proof of Theorem 1 and notations defined therein, note that  $O_{aux}$  can be written as:

$$O_{aux}(\mathbf{T}) = 1 - \sum_i \sum_{j \neq i} Q((\mathbf{p}_j - \mathbf{p}_i)\Sigma^{-1}(\mathbf{p}_j - \mathbf{p}_i)^T) P(H_i), \quad (31)$$

where  $Q(x)$  denotes the Q-function [3]. Now, suppose we wish to compute  $O_{aux}(\mathbf{T} \cup \{s_k\})$  for a sensor  $s_k$  whose observations have a mean of  $p_{ki}$  for hypothesis  $H_i$  and a variance is  $\sigma_k^2$ . Let us denote the argument of  $Q()$  in Eq (31) by  $\mathbf{q}_{ij}(\mathbf{T})$ . Then, we have the following recurrence relation:

$$\begin{aligned} O_{aux}(\mathbf{T} \cup \{s_k\}) &= 1 - \sum_i \sum_{j \neq i} Q(\mathbf{q}_{ij}(\mathbf{T} \cup \{s_k\})) P(H_i) \\ &= 1 - \sum_i \sum_{j \neq i} Q(\mathbf{q}_{ij}(\mathbf{T}) + \frac{(p_{ki} - p_{kj})^2}{\sigma_k^2}) P(H_i) \end{aligned}$$

We note that computing  $\mathbf{q}_{ij}(\mathbf{T})$  directly using Eq (31) takes  $O(B^2)$  time. However, we can compute  $\mathbf{p}_{ij}(\mathbf{T})$  incrementally by using the equation

$$\mathbf{q}_{ij}(\mathbf{T} \cup \{s_k\}) = \mathbf{q}_{ij}(\mathbf{T}) + \frac{(p_{ki} - p_{kj})^2}{\sigma_k^2}$$

in constant time. As computing the Q-function takes constant time, the above reduced the time complexity by a factor of  $O(B^2)$ .

## REFERENCES

- [1] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.
- [2] E. Seneta, "On the history of the strong law of large numbers and boole's inequality," *Historia Mathematica*, vol. 19, no. 1, pp. 24 – 39, 1992. doi: 10.1016/0315-0860(92)90053-E
- [3] M. Abramowitz, I. A. Stegun *et al.*, "Handbook of mathematical functions with formulas, graphs, and mathematical tables," 1972.