A Gaussian Process Based Technique of Efficient Sensor Selection for Transmitter Localization

Aritrik Ghosh
University of Calcutta
garitrik@gmail.com

Arani Bhattacharya
IIIT Delhi
arani@iiitd.ac.in

***Abstract*—Spectrum monitoring via crowdsourcing is a technique that promises to enable opportunistic spectrum access. Crowdsourcing aims to provide incentives to users to deploy a large number of cheap but potentially noisy sensors. The sensors all send their data to a fusion center, where typically some algorithms are used to remove the noise from the data. Such crowdsourced monitoring of spectrum has been shown to be feasible in practice in multiple studies. One of the key goals of such monitoring is to identify any users that are violating the protocols of accessing spectrum. While a number of crowdsourcing techniques to identify such violations have been proposed, a key challenge that remains is to minimize the cost of data consumption and energy of running the sensors. In this work, we propose sequential probing of sensors to accurately localize/identify such transmitters. We formulate this as a Gaussian Process multi-armed bandit problem, and use a widely known solution technique called Upper Confidence Bound to solve it. We next observe that such sequential probing incurs additional latency, and use batched selection of sensors in few rounds to reduce latency. We show that instead of naively selecting sensors in parallel batches, an intelligent technique of selecting sensors called Gaussian Process Adaptive Upper Confidence Bound (GP-AUCB) can lead to selection of sensors that can lead to more accurate localization. Finally, we show the tradeoff between accuracy of localization, latency incurred and number of selected sensors via simulations.**

I. Introduction

With the increasing cost of RF spectrum, there is a perceived need to monitor and protect it from unauthorized users. A key technique proposed for such spectrum patrolling is to utilize cheap crowdsourced sensors [1]–[4]. Such crowdsourced sensors have the advantage of being relatively easier to deploy as they have a relatively small form-factor. However, they also have the problem that they generate a huge amount of data, and the data generated is noisy. Thus, cleaning and processing this data to obtain inferences about the location or presence of transmitters requires sending it to a fusion center over a network. This increases the runtime cost of running such system, which if unchecked would increase the overhead of maintaining a tiered spectrum access system.

Prior works have limited the cost of such spectrum patrolling in a crowdsourced spectrum system by selecting the most relevant sensors. [1], [5]–[7]. For example, [1] uses a greedy feature selection technique called Maximum Relevance Minimum Redundance to select sensors. The work [5] proposes a framework based on submodularity to greedily select the most relevant sensors. POMES [6] considers the monetary incentive that needs to be paid to users, and selects a sensor only if the benefit obtained by using it is higher than the offered monetary incentive. Finally, [7] identifies the sensors that should be utilized when they can harvest their own energy.

A key common point among all the studies discussed above is that they all select the sensors a priori. In other words, the sensors are chosen before transmitter localization/identification. Alternatively, the decision about whether to probe the sensors is taken individually for each sensor [6], [7], which can lead to suboptimal results. On the other hand, a process of sequential selection of sensors, where the output of one selected sensor is used to guide the selection of subsequent sensors can lead to substantial improvement of accuracy at a lower cost.

However, designing an algorithm of sequential selection of sensors is non-trivial. The key reason is that the crowdsourced sensors are noisy in nature. A naive greedy approach that selects the sensor where the transmitter is most likely to be present might end up influenced by the noise and selecting a poor sensor. Note that in traditional sensor selection, a modified form of the greedy algorithm provides good performance due to the principle of submodularity [5], [8]. In sequential selection, on the other hand, there is an inherent tradeoff between *exploiting* the information given by the sensors, while also *exploring* sufficiently to ensure that the noisy sensors do not influence the selection too much.

A second challenge that comes up is that selecting such sensors sequentially can increase the latency of localization. This has potentially undesirable implications, as an unauthorized transmitter might potentially stop its transmission before it is localized [9]. To mitigate this problem, as a second step, we design a batched selection technique, where multiple sensors are selected in rounds. We show that by selecting a relatively few number of batches, it is possible to get accuracy very close to fully sequential selection, while incurring relatively small additional latency.

We formulate the problem of using the tradeoff between exploration and exploitation as a multi-armed bandit problem. A number of prior works have posed the problem of localization as one of Gaussian Process Regression [10], [11]. We borrow this approach of localization because it is known to scale to large areas, as well as lead to efficient solutions. Thus, our sensor selection strategies work by modeling the problem of sequential selection as Gaussian process multi-armed bandit

problem. We map the selection of sensors as a case of pulling the arms in a multi-armed bandit problem. This leads us to a widely used algorithm called Upper Confidence Bound (UCB) in selecting the sensors [12]. We show via simulation that our technique works much better in practice than the greedy algorithm.

We address the problem of latency involved in sequential sensor selection in the following way. Instead of sequentially selecting all the sensors, we select a batch of sensors together. We formulate this as the related problem of batched Gaussian Process multi-armed bandit problem [13], and use a modified form of UCB to select the sensors. Again, we show via simulation that the amount of accuracy lost in localization due to batches is relatively small.

We summarize our contributions as follows:

1) We first formulate the problem of sequential sensor selection to realize substantial cost savings while localizing an unauthorized transmitter as an instance of *Gaussian Process multi-armed bandit problem.*
2) We utilize the Upper Confidence Bound technique to solve the problem of sequential sensor selection.
3) We further reduce the latency involved in selecting sensors by selecting them in parallel batches. To do so, we use an algorithm called *Gaussian Process Adaptive Upper Confidence Bound*, which performs better than the standard selection in batches.
4) Finally, we study the tradeoffs between the three parameters, latency, cost and accuracy of localization using our designed techniques. Our evaluation shows that selecting sensors in parallel batches leads to increase in localization error due to inferior feedback. However, the adaptive upper confidence can mitigate some of this increase in localization error.

The rest of this paper is organized as follows. We first discuss related works in Section II. We next explain the system model in depth and the technique of sequential and batched sensor selection in Section III. We evaluate our techniques in Section IV, and conclude in Section V.

II. Related Work

Energy-efficient Spectrum Sensing: The rise of the Internet of Things with its demand of spectrum has led to the need for energy efficient spectrum sensing. There are broadly three types of works that try to make spectrum sensing energy-efficient. The first technique of improving energy-efficiency is to utilize energy harvesting [7], [14], [15], where energy from ambient sources is utilized by the spectrum sensors. The second technique is to utilize more efficient hardware like FPGA's [16] or embedding hardware to smartphones [4]. The third category of works deal with intelligent selection of channels and/or sensors for sensing. For example, [17] senses the channels that are more likely to be used with higher probability than the other more occupied channels. A similar technique is used by [18], where past history is utilized to determine the channels that should be scanned. A number of works focus on selecting the most relevant users

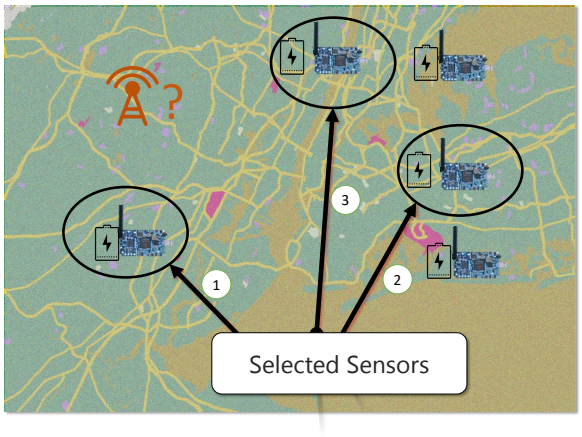

Fig. 1. An illustration of the system of sequential sensor selection to localize an unauthorized transmitter. The number denotes the batch in which the sensor is selected.

in an environment where a number of sensors are present [5], [19]. Our work builds on these techniques of sensor selection, but attempts to select sensors in a more intelligent way by selecting them in sequential fashion. The work [20] comes closest to our work. However, unlike our Gaussian Process Regression technique, it uses a hypothesis-based approach, which makes it harder to scale to larger number of sensors in practice.

Localization using Gaussian Processes: A number of works perform localization using Gaussian processes under different settings. For example, [10] showed that localization using Gaussian processes outperforms other techniques like k-nearest neighbors. Gaussian processes is the most common choice for fingerprint-based localization, using WiFi [11], low-powered wide area network (LPWAN) [21] and cellular networks [22]. Unlike our work, these works do not focus on selecting the most relevant sensors while performing localization.

Multi-armed bandit Problem in Gaussian Processes: Selecting sensors in a Gaussian Process is done using a greedy algorithm, as it provides near-optimal performance [23]. In this case, the greedy algorithm chooses the sensor with the highest mean. However, as shown in this study as well as our own experiments, the greedy algorithm performs poorly in case of sequential sensor selection with noisy sensors. Thus, the work [12] proposes an algorithm called *Gaussian Process Upper Confidence Bound* which selects the sensor with the highest value of the weighted sum of mean and standard deviation of each sensor. The batched version of this problem was studied in [13]. Our study models the problems of sequential and batched sensor selection as taking action in Gaussian processes, and then studies the performance tradeoffs for different batch sizes. A few recent works propose a different algorithm to solve this problem called parallel Thompson Sampling [24], but this is orthogonal to our approach and we do not consider it in this study.

III. Problem Formulation

We define the problem as follows. We are given an area, which may contain a stationary transmitter at an unknown location. A set of spectrum sensors $\mathbb{S}$ are deployed within this area. Selecting a sensor would immediately send its output to the fusion center, and the data becomes available with some delay. For convenience, we denote the set of sensors selected in round k by $\mathbb{T}_k$, and their output by $\mathbf{x}_k$. For convenience, we also denote the set of sensors selected **till** round k as $\mathbb{R}_k$, i.e.

$$\mathbb{R}_k = \mathbb{T}_1 \cup ... \cup \mathbb{T}_k.$$

Note that each of the sensors send the power received plus an additional noise to the fusion center. We assume that the noise of each sensor is Gaussian in nature, with the variance of a sensor being equal to ϵ.

The performance of localization is measured in terms of the estimate of the location of the transmitter. A point having the highest value of power is the actual location of the transmitter. In our setting, since the noise is Gaussian with known variance, the most accurate possible solution is given by the position $p(x_{\mathbb{S}})$ obtained using the observations from all possible sensors $x_{\mathbb{S}}$ [1]. Thus, we define the objective as minimizing the Euclidean distance between the estimate obtained using all sensors and the one obtained using the selected subset of sensors.

$$\textbf{Minimize } ||p(x_{\mathbb{R}k}) - p(x_{\mathbb{S}})||_2 \text{ subject to } |\mathbb{R}_k| \leq B, \quad (1)$$

where B is the maximum number of sensors that can be selected.

However, we note that the estimate obtained on selecting each sensor is obtained via Bayesian update. Since Bayesian update is known to be non-linear, the overall objective is also non-linear, which makes solving the above problem difficult. We therefore make an observation to make the problem linear. We note that if a transmitter is close to the sensor s, the output generated on average $E[x_s]$ is higher. Thus, we redefine our objective as that of maximizing the sum of average power f received across all sensors. Therefore, the objective or reward to the Gaussian Process multi-armed bandit becomes the expression:

$$\textbf{Maximize } \sum_{k=1}^{K} f(\mathbf{x}_k).$$

We call the best possible solution or the oracle solution $x*$ as the one which was possible *if the location of the transmitter was known*. We call the difference between the reward of the solution given by an algorithm and that of the oracle solution as the regret G:

$$G_k = \sum_{k=1}^{K} |f(x*_k) - f(x_k)| \quad (2)$$

[1]Note that as discussed in [25], in the presence of non-Gaussian noise, selecting all the sensors no longer gives the highest accuracy of localization. We provide the performance guarantees in terms of regret. Note that minimizing regret is equivalent to maximizing reward, so providing bounds on regret also leads to optimality results on the objective function.

We now study two variants of this problem. The first is the *sequential* variant, where the sensors are chosen sequentially, i.e. $|\mathbb{T}_k| = 1, \forall k$. The second is the *batched* variant, where the number of allowed rounds is smaller than the sensor budget, i.e. $|\mathbb{T}_k| \geq 1, \exists k$. In this case, our algorithm has to both decide the number of sensors to be selected in a round k, as well as the actual sensors to be selected.

A. Background of Gaussian Process

A Gaussian Process $GP(\mu(x), k(X, X'))$ is specified by its mean $\mu(x) = E[f(x)]$ and co-variance function $k(x, x^*) = E[(f(x) - \mu(x^*))(f(x^*) - \mu(x^*))]$, Where $E(.)$ is expectation. Consider a prior observation sensor set of size T from the entire sensor set D, and observe noisy signal sample $y_T = [y_1...y_T]^T$ at points $A_T = x_1...x_T$. Let $k_T(x) = [k(x_1, x)...k(x_T, x)]^T$ and K_T be the positive definite kernel matrix $[k(x, x^*)]_{x,x^* \in A_T}$. Then mean $\mu_T(x)$, co-variance $k(x, x^*)$ and variance $\sigma_T^2(x)$ can be mathematically expressed as follows:

$$\mu(x) = k_T(x)^T (K_t + \sigma^2 I)^{-1} y_T, \quad (3)$$

$$k_T(x, x^*) = k(x, x^*) - k_T(x)^T (K_T + \sigma^2 I) - 1 k_T(x^*), \quad (4)$$

$$\sigma_T^2 = k_T(x, x). \quad (5)$$

We have taken the Matérn kernel as the choice of positive definite kernel function, as prior studies like [10] have shown that it works well in practice. Let B_v be a modified Bessel function and l be a length scale parameter. Then Matérn kernel can be defined as follows:

$$k(x, x^*) = (2^{1-v}/\Gamma(v)) r^v B_v(r), r = ((2v)^{0.5}/l)) \parallel x - x^* \parallel, \quad (6)$$

v controls smoothness of the function. Matérn kernel provide a sub linear regret bound as mentioned in [12].

B. Solution for Sequential process

Maximising f requires us to chose points x_t so that the function f can be estimated well globally. Let us denote the covariance matrix by $K(A, A)$. Further, let H be the differential entropy of the probability distribution over the set of observations. Then, the entropy at a particular sensor measures the amount of uncertainty or the variance of the Gaussian process at a particular location. Thus, a good metric is the reduction in entropy if a sensor is selected, which is known as the mutual information. Mathematically, the conditional mutual information gained from each observation in y_A is given by:

$$I(y_A, f) = H(y_A) - H(y_A|f) = 0.5 log|I + \sigma_n^{-2}.K(A, A)|. \quad (7)$$

Modelling f as a sample form a GP has the major benefit that the predictive uncertainty can be used to guide exploration and exploitation.We chose the algorithm GP-UCB (Gaussian

Algorithm 1 The Gaussian Process Upper Confidence Bound algorithm (GP-UCB) used to select sensors sequentially

Require: Decision set D, GP prior mean μ_0, variance σ_0^2, kernel function(marten kernel) $k(.,.)$, Maximum Resource T.
for $t = 1, 2, ..T$ **do**
Choose $x_t \leftarrow argmax_{x \in D}[\mu_{t-1}(x) + \beta_t^{0.5}\sigma_{t-1}(x)$ ▷ $\beta_t = 2log(|D|t^2\pi^2/6\delta)$, $\delta \in (0,1)$
Sample $y_t = f(x_t) + \epsilon_t$
Perform Bayesian update to obtain μ_t and σ_t.
end for

Process Upper Confidence Bound) for selection of the sensor sets x_t at each round t. Let β_t be a domain specific time varying parameter to trade off exploration vs exploitation.$\beta_t = 2log(|D|t^2\pi^2/6\delta)$, $\delta \in (0,1)$. GP-UCB explores by sampling x with large variance and exploits by sampling x with large mean. The rule for choosing x_t is given by:

$$x_t = \arg\max_{x \in D}[\mu_{t-1}(x) + \beta_t^{0.5}\sigma_{t-1}(x)]. \quad (8)$$

It was shown by [12], that GP-UCB was able to bound regret with a quantified high probability. Thus, we choose it to solve this problem.

C. Batched Process

It is possible to select multiple sensors at each round by selecting the sensors in parallel. Introducing parallelism reduces the execution time. However, parallelism also comes with multiple overheads – (i) choosing multiple sensors take more time than choosing a single sensor, and (ii) the sensors are selected with inferior feedback. Thus, certain changes are needed in the basic GP-UCB to take into account these overheads.

We first model the time as follows. The total time taken consists of both the time taken to run the Bayesian update (t^{bay}) as well as average time to select each sensor (t^{sel}). We model this total time per round as follows.

The cost of execution till round n is given by the weighted sum of each individual cost:

$$C_n = 0.3 * \sum_{i=1}^{n}(t_i^{bay}) + 0.7 * \sum_{i=1}^{n}(t_i^{sel}). \quad (9)$$

This cost is averaged over multiple simulation in the same environment. Algorithm presented by [13] could be used to select B sensors at each time t, but parallelism comes with an price of inferior feedback. So certain changes are needed to be made in the sequential process to make parallel selection work in our scenario. Firstly let's assume that there is a mapping $fd : N \rightarrow \{N, 0\}$, this fd the feedback is such that $fd[t] \leq t - 1, \vee t \in N$, in the batched selection setting. $fd[t] = t - 1$ in the case of sequential selection and $fd[t] = 0$ when there is no feedback. In simple batched sensor selection setting $fd[t]_{batchsize=B} = \lfloor (t-1)/B \rfloor B$.

GP's predictive variance at time t only depends upon the $X_{t-1} = \{x_1, x_2, ...x_{t-1}\}$ and not on $y_1 : t - 1$. In our case the variance will depend upon the previous sensor which we have observe not the observed value. So in parallel case the posterior variance can be calculated similarly like the GP-UCB, even when the observation are not present. But, the mean depend upon the actual observation. So, to incorporate such changes one need to introduce a decision rule that sequentially chooses actions within the batch along with the information present. It is akin to running GP-UCB based on some hallucinated observation. The hallucination is done for those observation which were yet to get received. Hallucination is done by using most recent posterior mean for those observation. Regret of batched process is dependant on the batch size , but can be made independent by proper choice of initial data set. This can be done by using uncertainty sampling for T_{init} rounds to collect D_{init} sensor points and adding these back to posterior of the Gaussian process. We utilize the following lemma from [13] to provide a performance bound on the regret.

Lemma 1 *Suppose the kernel and T_{init} and bath size $B \geq 2$ and $0 < \delta < 1$ are fixed. Let R_t be the cumulative regret at round T of GP-BUCB algorithm , initialised by uncertainty sampling GP-BUCB algorithm, which ignores feedback for the first T_{init} rounds. Then there exists a constant C^* independent of batch size B such that*

$$Pr(R_T \leq C^* R_T^{seq} + 2\| f \|_\infty T_{init}, \vee T \geq 1) \geq 1 - \delta. \quad (10)$$

It can be summarised from Lemma 1, the regret is bounded by sum of two terms. 1st term is responsible for running the algorithm after initialisation and second term is for the initialisation phase. Even though the second term is dependent on batch size B because T_{init} depends on B, it's influence is limited and the sum is particularly dependent upon the $C^* R^{seq}$ where R^{seq} is the regret of GP-UCB, so for the choice of particular kernel (Linear, Marten or RBF) and T_{init} as mentioned in Table 1 of [13], regret bound of GP-BUCB is independent of B and it is worse than GP-UCB. The choice of β_t will be same as that of GP-UCB.

Algorithm 2 Uncertainty Sampling

Require: Decision set D. GP prior mean μ_0, variance σ_0^2, kernel function(marten kernel) $k(.,.)$.
for $t = 1, 2, ..T$ **do**
$x_t \leftarrow argmax_{x \in D}\ \sigma_{t-1}(x)$
Update $\sigma_t(x)$
end for

D. Adaptive Batch

Deriving from GP-BUCB it is also possible for the algorithm to control the flow of parallelism. Reference [13] suggests an algorithm called GP-AUCB (Gaussian process adaptive upper control bound) Local which can control the parallelism. In our problem both the number of iteration and number of parallel process comes with an cost. So using such adaptive process to control the batch size and reducing the

Algorithm 3 GP-BUCB

Require: Decision set D, GP prior mean μ_0, variance σ_0^2, kernel function(marten kernel) $k(.,.)$, feedback map $fd[.]$, number of rounds T.
for $t = 1, 2, ...T$ **do**
$x_t = argmax_{x \in D}[\mu_{fd[t]}(x) + \beta_t^{0.5}\sigma_{t-1}(x)]$ ▷ $\beta_t = 2log(|D|t^2\pi^2/6\delta)$, $\delta \in (0,1)$
Calculate $\sigma_t(.)$
if $fd[t] < fd[t+1]$ **then**
$y_t^* = f(x_t^*) + \epsilon_t^*$ for $t^* \in (fd[t]+1, ...fd[t+1])$
Obtain $\mu_{fd[t+1]}(.)$ ▷ The Hallucinated mean
end if
end for

Algorithm 4 GP-AUCB Local

Require: Decision set D,GP prior mean μ_0, variance σ_0^2, kernel function $k(.,.)$, feedback map $fd[.]$, constant C, maximum batch size B_{max}.
Set $fd[t^*] = 0, \vee t^* \geq 1$.
for $t = 1, 2...T$ **do**
if $t - fd[t] > B_{max} \parallel \exists x \in D : \sigma_{fd[t]}(x)/\sigma_{t-1}(x) > exp(C)$ **then** ▷ $\sigma_{fd}[t]$ is hallucinated deviation
$y_{t^*} = f(x_{t^*}) + \epsilon_{t^*}$ for $t^* \in (fd[t-1], ..., t-1)$
Obtain $\mu_{t-1}(.)$
Set $fd[t^*] = t-1, \vee t^* \geq t$
end if
Choose $x_t = argmax_{x \in D}[\mu_{fd[t]}(x) + \beta_t^{0.5}\sigma_{t-1}(x)]$ ▷ $\beta_t = 2log(|D|t^2\pi^2/6\delta)$, $\delta \in (0,1)$
Calculate $\sigma_t(.)$
end for

total number of iteration compared to GP-UCB greatly benefits our cause. The key behind GP-AUCB algorithm is choosing the feedback $fd[t]$ online (formally defined in Algorithm 4). This practice helps to limit the amount of information hallucinated within a batch. Maximum batch size B_{max} along with a constant C is used as stopping criteria to control the parallelism. The choice of C is abstract and it should maintain the condition $I(f; y_{fd[t]+1:t-1}|y_{1:fd[t]}) \leq C, \vee x \in D, \vee t > 1$. Maintaining this condition ensures the confidence intervals which is used to select actions are faithful to the true posterior. In other words, $\sigma_{t-1}(x)$ does not become too small compared to the variance $\sigma_{fd[t]}(x)$ of hallucinated observation.

The choice of β_t is identical to that of GP-UCB. The algorithm makes sure that the batch size does not exceed the maximum batch size criteria. It also runs on a tighter regret bound than GP-UCB since the constant C only needs to exceed the local information gain.

As the size of the solution set is finite, finding x_t is feasible and its performance is expected to be stable in the real life settings. We confirm that this is true in the evaluation.

IV. Evaluation

Evaluation Setting: For the simulation we have considered an grid represents an area of $4sqkm$ with 40×40 grid points.

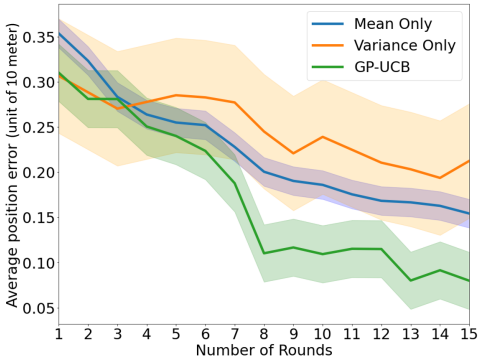

Fig. 2. Confidence interval of average position error of predicted transmitter location (with unit of 10 meter) of GP-UCB,Mean only and Variance only update rule in S-ALL setting.

Then we interpolated the power so obtained to grid points of dimensions $10,000 \times 10,000$. Thus, each grid point represents a single cell of dimension $0.04m^2$. We assume that sensors are stationary and their locations are known. Then the transmitter is placed randomly within the grid, and its location is unknown to us. The signal power is simulated using the tool SPLAT! [26] which uses Longley-Rice model. Every sensor reports the power mixed with some zero-mean Gaussian noise. The variance of this Gaussian noise is taken as 10% of the received signal. The outputs of the selected sensors are used to compute the posterior of Gaussian Process for predicting the position of the transmitter. The simulation results are produced by taking an average over 20 unknown transmitter location for each case.

We show two different versions of the results to show performance. The first technique of evaluation, which we call as S-ALL, utilizes all the sensors for evaluation of f. The second technique, which we call as S-BEST, utilizes the sensor that reports the highest power for evaluation of f. S-BEST is shown typically to compare the performance for different number of rounds with the purely sequential GP-UCB technique. We evaluate in both the evaluation settings.

A. Sequential Selection Process

Figure 2 compares the average position error over 20 arbitrary location of the transmitter, when GP-UCB is used along with two other update rules, one which only considers mean ("Mean-only") and another one which only considers variance ("Variance-only"). We note that the Variance-only method is equivalent to a priori (offline) selection of sensors, as selecting the sensors with highest mutual information or variance is equivalent to selecting the best sensors a priori. It can also be seen as a case of pure exploration, since the sensors are chosen without any utilization of their outputs. The "Mean-only" technique, on the other hand, can be seen as a case of pure exploitation, since the sensors are chosen only at the point of highest prior beliefs.

When the number of sensors were less than 5, all 3 algorithm performed similarly but as the number as senors increased to 10 and then to 15 the GP-UCB consistently outperforms "Mean-Only" and "Variance-Only" by an average

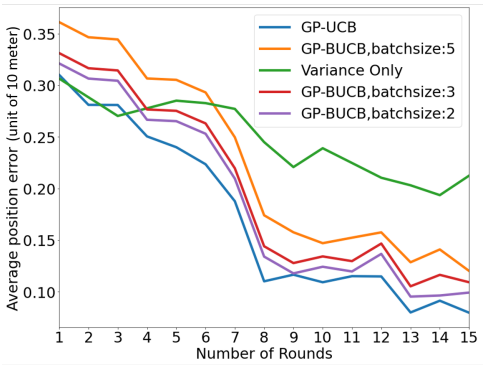

Fig. 3. Average position error of predicted transmitter location (with unit of 10 meter) of GP-UCB , "Variance only" , GP-BUCB with batch size 2, 3 and 5 across 20 rounds plot in S-BEST setting.

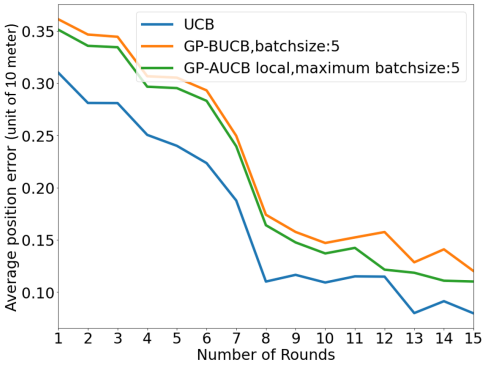

Fig. 4. Average position error of predicted transmitter location (with unit of 10 meter) of GP-UCB . GP-BUCB and GP-AUCB local across 20 rounds plot in S-BEST setting.

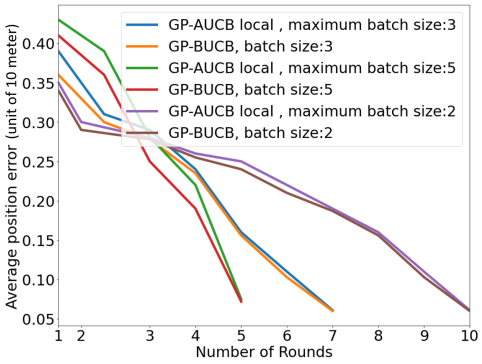

Fig. 5. Average position error of predicted transmitter location (with unit of 10 meter) vs number of rounds plot of GP-BUCB and GP-AUCB local in S-ALL setting.

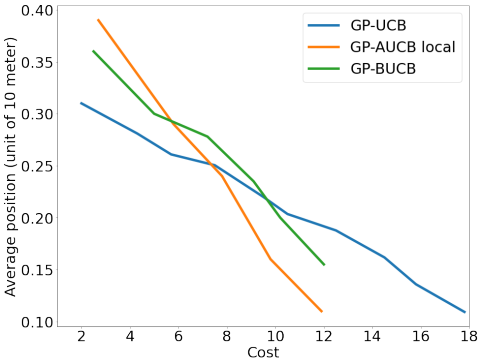

Fig. 6. Average position error of predicted transmitter location (with unit of 10 meter) vs cost of GP-UCB ,GP-BUCB and GP-AUCB local in S-ALL setting.Here the result is obtained by varying batch size and maximum batch size of GP-BUCB and GP-AUCB local respectively form 2 to 5 in both the cases.

of 27% and 37.7% respectively. Moreover, the "Variance-Only" method has a very high confidence interval, as its selection is unguided in nature and can thus lead to very poor results occasionally. Clearly GP-UCB outperforms the other two algorithm for predicting the location of the transmitter. This clearly shows that GP-UCB is a good technique of using a mix of exploitation and exploration.

B. Batched Selection Process

1) Accuracy-Round Tradeoff: Figure 3 compares the result of batched process (GP-BUCB) with the sequential GP-UCB and "Variance-Only" method in S-BEST setting. We observe that GP-BUCB gives similar result for batch sizes of 2, 3 and 5 and performs better than "Variance only" by an average of 22.2%.The GP-UCB works well and on average it outperforms GP-BUCB by 15.7% when used to reduce the average localisation error. We note that the error consistently increases with an increase in batch size, which indicates that reducing latency comes with a cost.

2) Performance of GP-AUCB: Figure 4 compares the result of all three techniques GP-UCB, GP-AUCB local and GP-BUCB when best choice from each round of the batched processes is taken. In this case even though GP-AUCB local preforms better than GP-BUCB by 9% , it is still not as good as GP-UCB. Moreover, the localization error in case of GP-AUCB is more stable than GP-BUCB, as GP-BUCB shows an increase in error when we move from selecting 12 to 13 sensors. This shows that GP-AUCB is able to mitigate the amount of increase in localization error seen in GP-BUCB, due to the reduction in number of rounds (and thus amount of latency). We acknowledge that there is still some amount of performance gap between GP-UCB and GP-AUCB, indicating that better algorithms might be possible.

Figure 5 shows when using the entire sensor set form the batched processes for predicting location of the sensor. For batch sizes 2, 3 and 5 both the batched process were able to produce similar prediction performance like GP-UCB and it took them less than half number of rounds compared to GP-UCB.

Finally with the help of the cost defined previously in Section IV.B, we can observe from Figure 6 that GP-UCB works well in the low cost scenarios. When the cost is increased for better localisation, in terms of latency GP-AUCB local performs better than GP-UCB by 33% and in terms of accuracy GP-AUCB local performs 24% better than GP-BUCB.

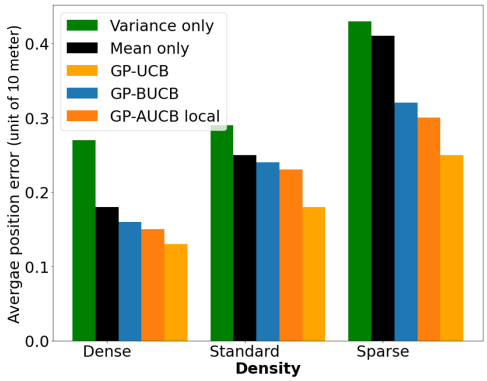

Fig. 7. Comparison of localization accuracy for different density of sensors in the environment, with a fixed limit of 7 sensors in S-ALL setting.

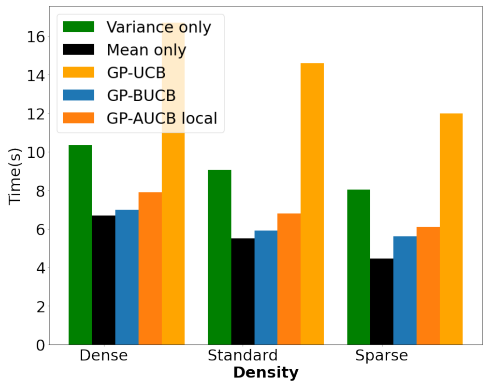

Fig. 8. Time comparison for different density of sensors in the environment, with a fixed limit of 7 sensors. Dense represents a case with a total with 500 sensors, standard represents a case with 200 sensors and sparse a case with 50 sensors.

3) Performance with Varying Sensor Density:: Figure 7 shows the average error in localization for varying sensor density (i.e., number of available sensors in the simulation environment) with a fixed limit of 7 sensors. Sensor set size of 50, 200 and 400 chosen for sparse , standard and dense case respectively. Sparse case provides worst result because of lack of sensor choice and average proximity from the sensor from the transmitter. GP-UCB shows a gradual increase in the performance from moving from spare to standard and standard to dense by 28% and 26% respectively. Performance of GP-BUCB and GP-AUCB local is quite similar, but GP-AUCB local still performs slightly better than GP-BUCB.

4) Scalability: We run all the algorithms on Google Colab, with a single-core Intel(R) Xeon(R) CPU @ 2.00GHz having 12 GB of RAM. Figure 8 shows the amount of time each algorithm takes to select 7 sensors, for sparse, standard and dense sensor density. GP-UCB took the longest in each case whereas both the batched processes and "Variance only" method took similar time. This is because evaluation time for GP-UCB update rule is higher compared to "Variance only" and "Mean only" method. But both the batched process evaluate the update rule in parallel which reduces the number of iterations to find similar number of sensors compared to GP-UCB.

V. Conclusion

In this paper, we showed a technique of sequential selection of sensors to localize an unauthorized transmitter. We first showed that sequential sensor selection can outperform traditional sensor selection techniques, though at the cost of higher latency. We map this to the Gaussian Process multi-armed bandit problem, and by leveraging techniques proposed in existing literature, we utilized the Gaussian Process Upper Confidence Bound (GP-UCB) to solve it. To reduce the latency, we then propose to select the sensors in parallel batches. We first showed that using a simple technique called Gaussian Process Batched Upper Confidence Bound (GP-BUCB) leads to substantial increase in localization error. We thus utilize a more intelligent algorithm called GP-AUCB (Gaussian Process Adaptive Upper Confidence Bound) Local to reduce the increase in localization error. We perform large-scale simulation to validate our approach and show that our approach scales to large number of sensors.

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*, vol. 6, no. 1, pp. 271–281, 2020. DOI: 10.1109/TCCN.2019.2939793

[2] A. Nika, Z. Li, Y. Zhu, Y. Zhu, 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 CD-ROM*, ser. SenSys '16. New York, NY, USA: Association for Computing Machinery, 2016. DOI: 10.1145/2994551.2994557. ISBN 9781450342636 p. 96–108. [Online]. Available: https://doi.org/10.1145/2994551.2994557

[3] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. Van den Bergh, H. Cordobes, D. Giustiniano, S. Pollin, and V. Lenders, "Electrosense: Open and big spectrum data," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 210–217, 2018. DOI: 10.1109/MCOM.2017.1700200

[4] P. Smith, A. Luong, S. Sarkar, H. Singh, A. Singh, N. Patwari, S. K. Kasera, and K. Derr, "A novel software defined radio for practical, mobile crowd-sourced spectrum sensing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021. DOI: 10.1109/TMC.2021.3107409

[5] A. Bhattacharya, C. Zhan, H. Gupta, S. R. Das, and P. M. Djuric, "Selection of sensors for efficient transmitter localization," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020. DOI: 10.1109/INFOCOM41043.2020.9155230 pp. 2410–2419.

[6] S. Bayhan, G. Gür, and A. Zubow, "Pomes: Profit-maximizing sensor selection for crowd-sensed spectrum discovery," in *Cognitive Radio-Oriented Wireless Networks*, A. Kliks, P. Kryszkiewicz, F. Bader, D. Triantafyllopoulou, C. E. Caicedo, A. Sezgin, N. Dimitriou, and M. Sybis, Eds. Cham: Springer International Publishing, 2019. DOI: 10.1007/978-3-030-25748-4_1 pp. 3–16.

[7] A. F. Tayel, S. I. Rabia, A. H. A. El-Malek, and A. M. Abdelrazek, "An optimal policy for hybrid channel access in cognitive radio networks with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 253–11 265, 2020. DOI: 10.1109/TVT.2020.3008671

[8] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017. DOI: 10.1109/INFOCOM.2017.8057113 pp. 1–9.

[9] A. Nika, Z. Zhang, B. Y. Zhao, and H. Zheng, "Toward practical spectrum permits," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 1, pp. 112–122, 2017. DOI: 10.1109/TCCN.2017.2656127

[10] A. Bekkali, T. Masuo, T. Tominaga, N. Nakamoto, and H. Ban, "Gaussian processes for learning-based indoor localization," in *2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 2011. DOI: 10.1109/ICSPCC.2011.6061737 pp. 1–6.

[11] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, "Locating user equipments and access points using rssi fingerprints: A gaussian process approach," in *2016 IEEE International Conference on Communications (ICC)*, 2016. DOI: 10.1109/ICC.2016.7511152 pp. 1–6.

[12] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012. DOI: 10.1109/TIT.2011.2182033

[13] T. Desautels, A. Krause, and J. W. Burdick, "Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization," *Journal of Machine Learning Research*, vol. 15, no. 119, pp. 4053–4103, 2014. [Online]. Available: http://jmlr.org/papers/v15/desautels14a.html

[14] D. A. U. Villalonga, J. T. Gómez, and M. J. F.-G. García, "Optimal sensing policy for energy harvesting cognitive radio systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3826–3838, 2020. DOI: 10.1109/TWC.2020.2978818

[15] A. F. Tayel, S. I. Rabia, A. H. A. El-Malek, and A. M. Abdelrazek, "An optimal policy for hybrid channel access in cognitive radio networks with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 253–11 265, 2020. DOI: 10.1109/TVT.2020.3008671

[16] A. Bhattacharya, H. Chen, P. Milder, and S. R. Das, "Quantifying energy and latency improvements of fpga-based sensors for low-cost spectrum monitoring," in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2018. DOI: 10.1109/DySPAN.2018.8610459 pp. 1–5.

[17] N.-N. Dao, W. Na, A.-T. Tran, D. N. Nguyen, and S. Cho, "Energy-efficient spectrum sensing for iot devices," *IEEE Systems Journal*, vol. 15, no. 1, pp. 1077–1085, 2021. DOI: 10.1109/JSYST.2020.2986030

[18] T. S. Syed and G. A. Safdar, "History-assisted energy-efficient spectrum sensing for infrastructure-based cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2462–2473, 2017. DOI: 10.1109/TVT.2016.2585763

[19] S. Althunibat, R. Palacios, and F. Granelli, "Energy-efficient spectrum sensing in cognitive radio networks by coordinated reduction of the sensing users," in *2012 IEEE International Conference on Communications (ICC)*, 2012. DOI: 10.1109/ICC.2012.6363804 pp. 1399–1404.

[20] A. Bhattacharya, A. Maji, J. P. V. Champati, and J. Gross, "Fast and efficient online selection of sensors for transmitter localization," in *2022 International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2022.

[21] Y. Li, Z. He, Y. Li, H. Xu, L. Pei, and Y. Zhang, "Towards location enhanced iot: Characterization of lora signal for wide area localization," in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 2018. DOI: 10.1109/UPINLBS.2018.8559844 pp. 1–7.

[22] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "Gpps: A gaussian process positioning system for cellular networks," *Advances in Neural Information Processing Systems*, vol. 16, pp. 579–586, 2003.

[23] 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, no. 2, 2008.

[24] K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Poczos, "Parallelised bayesian optimisation via thompson sampling," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Storkey and F. Perez-Cruz, Eds., vol. 84. PMLR, 09–11 Apr 2018, pp. 133–142. [Online]. Available: https://proceedings.mlr.press/v84/kandasamy18a.html

[25] K.-H. Lam, C.-C. Cheung, and W.-C. Lee, "Rssi-based lora localization systems for large-scale indoor and outdoor environments," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11 778–11 791, 2019. DOI: 10.1109/TVT.2019.2940272

[26] S. Kasampalis, P. I. Lazaridis, Z. D. Zaharis, A. Bizopoulos, S. Zettas, and J. Cosmas, "Comparison of longley-rice, itm and itwom propagation models for dtv and fm broadcasting," in *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2013, pp. 1–6.