Multi-Agent Active Search and Rescue

Ramina Ghods
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
rghods@cs.cmu.edu

Arundhati Banerjee
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
arundhat@cs.cmu.edu

William J. Durkin
School of Earth Sciences
Ohio State University
Columbus, OH 43210
durkin.98@osu.edu

Jeff Schneider
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
jeff4@cs.cmu.edu

Abstract

In this paper, we propose a parallel active learning algorithm called SPATS (sparse parallel asynchronous Thompson sampling) that using multiple ground robots (agents) efficiently locates survivors of disasters. Unlike existing algorithms, SPATS is a practical algorithm that takes into account sparsity, lack of reliable communication to a central unit and sensing action constraints.

1 Introduction

Active search and rescue defines the problem of efficiently locating rescue mission targets in an unknown environment by interactively collecting data [6, 7]. Most existing active search algorithms are developed for a single agent and are not extendable to multi agent scenarios. As an example, [6] uses information greedy approaches to decide on best sensing actions for its agent. If we were to use multiple agents for this info-greedy method, all agents would make the same exact decision at each time step wasting resources of other agents. For other active search algorithms that are extendable to multi agent scenarios, they usually need a central control system to decide on the sensing actions and movements of all agents. However, in rescue missions agents generally lack a reliable communication channel to each other and more importantly to a central controller. Another consideration of this paper is a realistic constraint on the sensing actions. Inspired by search robots, we assume that each agent can only sense its immediate vicinity. In other words, each agent senses a triangular area in immediate camera view of the space at each time step. We furthermore model noise in our observations in accordance to the distance of the objects from the agent's position.

2 Problem and Motivation

Our goal in active search is to efficiently search for targets in an unknown environment by actively taking sensing actions given all the observations thus far. This can be thought of as an active learning problem setting (referred to as "Design of Experiment" in statistical literature [9]). In particular, looking at Figure 1, let us assume the gridded yellow area depicts an area of interest for the rescue mission and the marks "X" show the location of the rescue subjects. Our goal is to send agents ground robots to sense around and search for the rescue subjects without human interaction or any communication with a central unit and with as few measurements as possible. In other words, agents unreliably communicate their measurements with each other but there is no central unit to dictate

NeurIPS 2020 3rd Robot Learning Workshop: Grounding Machine Learning Development in the Real World.

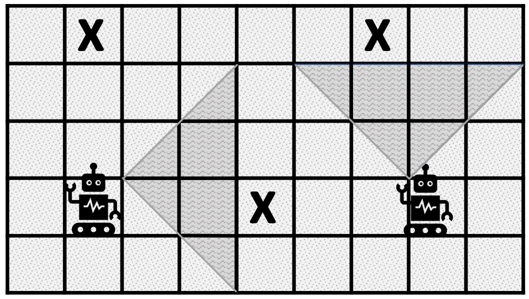


Figure 1: Robots are locating the objects of interest by searching the environment with its camera.

actions to the agents. Instead, each agent must be equipped with an intelligent data acquisition procedure that uses the available measurements from itself and the surrounding agents to decide on its next action, i.e. where to sense next.

While these assumptions are practical in a search and rescue application, existing search algorithms are generally active learning algorithms that focus on single agent scenarios and fail under multi-agent assumptions. This is because such algorithms are in general in batch mode and hence require a central unit to coordinate between agents at all times and pick a batch of sensing actions for all agents at a time (e.g. [1, 2]). In contrast, our active search problem has no central unit.

Mathematical Formulation: We describe the mission environment with a matrix B where we assume that we have no knowledge on its prior distribution other than knowing it is sparse. A sparse signal has only a handful of nonzero elements and is zero elsewhere. We assume he number of these nonzero elements are small but are unknown to the agents. Such assumption is very sensible for example in an earthquake survival mission where number of survivors are small (sparse) but unknown. For each agent, we can write the sensing operation as follows:

$$y_t = X_t\beta + n_t, n_t \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

Here, β with length N is the flattened version of matrix B, matrix X_t describes the sensing matrix at time t (grey triangles representing camera view point), y_t is the observation vector and n_t is its corresponding observation noise at time t. Let us assume the overall number of measurements available to all agents are T. Our objective is to estimate the sparse vector β with as few measurements T as possible. Each time step t, we choose a sensing action X_t given the data sequence $D_{t-1} = \{(X_1, y_1), \ldots, (X_{t-1}, y_{t-1})\}$.

3 Background and Related Work

In robotics, methods that deal with active search generally aim at autonomously building topological (identify obstacles and clearways) and/or spatial maps of a region. Our active search problem differs from topological mapping techniques such as SLAM [4] and can be most closely related to spatial mapping. For example, [7] identifies strong signals in environments with background information using trajectory planning with confidence intervals; but, unlike our problem setting, their algorithm is developed for a single agent performing point sensing observations. [5] formulates a multi agent game theoretic approach to coordinate unmanned aerial vehicles for cooperative search. However, they require the actions of neighboring agents for optimal action selection which we find impractical.

4 Our Proposed Method

To combat the multi agent issue, we propose Thompson Sampling (TS). TS is an exploration-exploitation algorithm applicable to active learning problems and originally introduced for clinical trials on two groups in [10]. Using TS properties as analyzed in [3], we can use TS to develop asynchronous parallel active learning method where each agent can make independent and intelligent decision on the next sensing action given the available measurements. Furthermore, [3] shows that such asynchronous TS algorithm outperforms all existing parallel Bayesian Optimization methods.

The TS approach is provided in Algorithm 1. In short, the algorithm consists of T steps. In each step, we in turn 1) Sample $\beta^\star$ from posterior distribution of β, 2) optimize for a sensing action X_t using a reward function that assumes the sample from step 1 is the true state of the world 3) using the sensing action from step 2 make a new observation and add it to the list of all available measurements.

4.1 Challenges of Thompson Sampling

Unfortunately, implementing TS with a sparse prior on vector β leads to poor performance that is on par with a point-wise algorithm that exhaustively searches all locations one at a time. Next, we will combat this challenge in 2 steps.

Step 1. Detecting the problem: To combat this challenge, we first need to identify the problem. We can associate this poor performance with one of the failure modes of TS discussed in Sec. 8.2 of the tutorial by [8]. According to this tutorial, TS faces a dilemma when solving certain kinds of active learning problems. One such scenario are problems that require a careful assessment of information gain. In general, by optimizing the expected reward, TS always restricts its actions to those that have a chance in being optimal. However, in certain active learning problems such as ours suboptimal actions can carry additional information regarding the parameter of interest. In particular, since the sample $\beta^\star$ is sparse, TS assuming this sample is the true state of the world would pick sparse sensing actions. However, if the algorithm had picked larger sensing actions like those in Figure 1, it could have eliminated a larger portion of the area as possible rescue subject locations.

Algorithm 1 Thompson Sampling

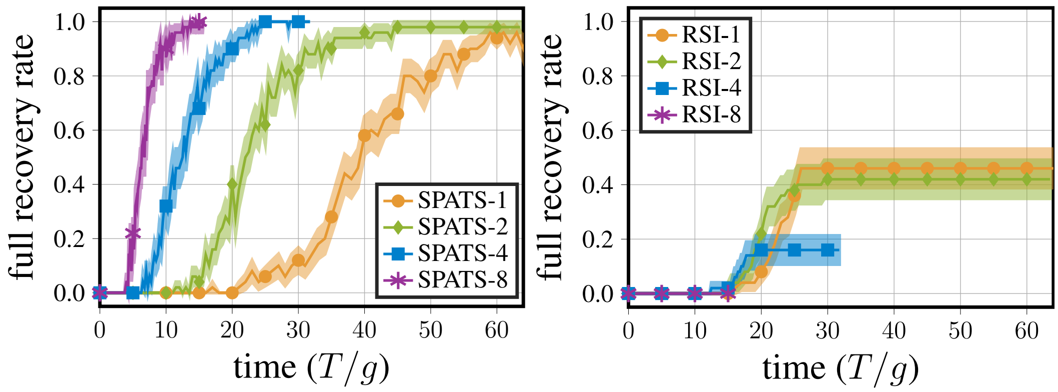
Assume: prior $\beta \sim p_0$ and likelihood $p(y_t|x_t, \beta)$
For $t = 1, ..., T$
  Sample $\beta^\star \sim p(\beta|D_{t-1})$ *Posterior Sampling*
  Select $x_t = \arg\max_x \text{Reward}(\beta^\star, D_{t-1}, x)$ *Design*
  Observe y_t given action x_t
  $D_t = D_{t-1} \cup (x_t, y_t)$

Step 2. Proposing a solution: To overcome this issue, we propose making an assumption on the prior distribution that the neighboring entries of the sparse vector β are spatially correlated, i.e. β is block sparse. Such spatial correlation creates the most compatible results to the region sensing constraint which only approves sensing actions with a single triangular non-zero block of sensors. Furthermore, we expect block sparsity to introduce exploration ability while also keeping sparsity a useful information in the recovery process. In particular, by gradually reducing the length of the blocks from a large value, we gently trade exploration with exploitation capability over time. I call this algorithm SPATS (short for Sparse Parallel Asynchronous Thompson Sampling) as given in Algorithm 2 for M agents. The derivation of the posterior and Reward of Algorithm 2 using a block sparse prior have been excluded here for sake of brevity. For this derivations, we use the block sparse prior and estimator proposed in [11].

Algorithm 2 SPATS: Sparse Parallel Asynchronous Thompson Sampling

---

Assume: Sensing model 1, M agents
Set: $D_0 = \varnothing$, block-sparse signal β with block length $L = N/M$
For $t = 1, ..., T$
  Wait for an agent to finish
  For the free agent:
    Sample $\beta^\star \sim p(\beta|D_{t-1}, \gamma, B)$
    Select $x_t = \arg\max_x \text{Reward}(\beta^\star, D_{t-1}, x)$
    Observe y_t given action x_t
    $D_t = D_{t-1} \cup (x_t, y_t)$
    if $t \,\%\, M = 0$ **then** $L = L/2$



(a) SPATS (b) RSI

Figure 2: Full recovery rate of SPATS and RSI for 1, 2, 4 and 8 agents for sparsity rate $k = 5$

5 Results and contributions

5.1 Results

We now compare the performance of our proposed SPATS algorithm against the information-theoretic approach called RSI algorithm proposed in [6] extended to multi-agent scenario. Throughout the experiments, we focus on locating 5 targets in an environment with grid size of $N = 8 \times 16$ and noise variance of $\sigma^2 = 1$. We then vary the number of measurements T and compare the mean and standard error of the full recovery rate over 50 random trials. The full recovery rate is defined as the rate at which an algorithm correctly recovers the entire vector β over the random trials.

Figure 2 shows the performance of SPATS versus RSI with 1, 2, 4 and 8 agents. Here, SPATS significantly outperforms RSI because SPATS is carefully designed to use randomness from Thompson Sampling in its reward function such that multiplying the number of agents by M would multiply its full recovery rate by a factor of M (as evident in Figure 2a). However, without any randomness in RSI's reward function, multiplying the number of agents will not improve RSI's performance as all agents will be repeating the same sensing action at a given time on par with a single agent setting.

5.2 Contributions

- We proposed a novel multi-agent active search algorithm called SPATS for search and rescue missions that outperforms existing algorithms.
- SPATS actively locates sparse targets in an unknown Environment using multiple agents that asynchronously make independent data-collection decisions without the presence of a central controller.
- SPATS is a completely nonparametric algorithm and does not need to know the number of targets.
- We showed how sparsity in its nature limits the exploration factor in active learning and we propose block sparsity to tackle this problem.
- To demonstrate the efficacy of SPATS, we apply it to synthetic and photo-realistic game environments in an asynchronous multi-agent setting.
- Besides search and rescue missions, SPATS can be applied to any multi-agent active learning algorithm with sparsity constraint such as active learning applications of detecting gas leak, pollution sources, remote sensing, multi-armed bandits, mobile sensor networks and adaptive compressive sensing.

References

[1] Azimi, J., Fern, A., and Fern, X. Z. , "Batch Bayesian optimization via simulation matching", *Advances in Neural Information Processing Systems* , pp. 109–117, 2010.

[2] Gu, Q., Zhang, T., and Han, J. , “Batch-mode active learning via error bound minimization”, *Uncertainty in Artificial Intelligence*, pp. 300–309, 2014.

[3] Kandasamy, K., Krishnamurthy, A., Schneider, J., and Poczos, B. , “Asynchronous parallel Bayesian optimization via Thompson sampling”, *arXiv preprint arXiv:1705.09236* , 2017.

[4] Leonard, J. J. and Durrant-Whyte, H. F. , “Simultaneous map building and localization for an autonomous mobile robot”, *Intelligent RObots and Systems* , volume 3, pp. 1442–1447, 1991.

[5] Li, P. and Duan, H. , “A potential game approach to multiple UAV cooperative search and surveillance”, *Aerospace Science and Technology*, 68:403–415, 2017.

[6] Ma, Y., Garnett, R., and Schneider, J. ,“ Active search for sparse signals with region sensing,” *Thirty-First AAAI Conference on Artificial Intelligence* , 2017.

[7] Rolf, E., Fridovich-Keil, D., Simchowitz, M., Recht, B., and Tomlin, C. , “A successive-elimination approach to adaptive robotic sensing”, arxiv e-prints, 2018.

[8] Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., Wen, Z., “ A tutorial on Thompson sampling”, *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018.

[9] Settles, B. , “Active learning literature survey” *Computer Sciences Technical Report 1648, University of Wisconsin-Madison*, 2009.

[10] Thompson, W. R. , “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples”, *Biometrika* , 25(3/4):285–294, 1933.

[11] Zhang, Z. and Rao, B. D. , “Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning”, *IEEE Journal of Selected Topics in Signal Processing*, 5(5):912–926, 2011.