Comparison of Bayesian Search Algorithms

Drew Ellison

August 4, 2014

1 Objective

Integrated surveillance and reconnaissance (ISAR) missions are an important application class for co- operative networks of unmanned aerial vehicles (UAVs), which must provide timely information about adversarial activities, environmental conditions, and friendly asset status to support coordinated dynamic decision-making. To improve the robustness and performance of such systems in urban environments, the AFRL Munitions Directorate seeks to develop formal online estimation and planning strategies for conducting probabilistic target search using MAVs (such as quadcopters) in urban indoor/outdoor envi- ronments.

The development of robust autonomous target search strategies for quadcopters in urban environments leads to several key challenges. For instance, it is not immediately obvious how autonomous planning algorithms should best balance search objectives (information gathering) with operational limitations on battery life, sensing range, processing power, and communications. Furthermore, many search planning techniques based on discretization of search space, which scales poorly and is computationally expensive in large environments with many obstacles. These issues motivate the development of information-driven continuous planning techniques for quadcopters.

2 Research

2.1 Problem Outline

Before attempting to establish implementations of continuous, information driven search algorithms, it will be useful to assess the strengths and weaknesses of current algorithms that rely on the discretization of the search space. Chung et al. [1] describe several strategies that this research seeks to recreate and build upon for deployment on a real autonomous quadrotor. These strategies are listed below.

At this point, basic implementations of each of these algorithms have been created. These implementations do not explicitly include obstacle avoidance; however, they implicitly "know" that a grid cell containing an obstacle has zero probability of containing the target through the definition of the initial probability distribution. Explicit obstacle avoidance will be implemented in the next version of these algorithms.

After this has been achieved, these algorithms will be compared ina numerical simulation experiment. This experiment will assess each of these algorithms in more realistic scenarios in which the search grid-size is sufficiently large, various obstacle configurations will be used, and different prior probability distributions will be explored including uniform, Gaussian, and Gaussian mixture distributions.

Additionally, each algorithms performance will be measured with respect to several "global" search parameters. These parameters include the characteristics of the sensor being used (i.e. probability of a missed/false detection), grid size, and obstacle configuration. Sensitivities to algorithm-specific parameters will also be explored, e.g. window-sizes for lookahead-based optimal search strategies.

Several metrics will be used for the comparison of performance, with the main metric being average time until detection. Other metrics may include the aggregate belief probability (since these algorithms account for the possibility that the target is not in the search grid at all) and the variance in the time until detection.

2.2 Problem Setup

2.2.1 Grid and Sensor

The task of searching for a stationary target in a given search space can be solved by breaking the search space into discrete cells. The target cell location will be denoted x_T and the search space, $\mathcal{A}$, will be divided into $|\mathcal{A}|$ cells. In this implementation, the absence of the target from the search area will be allowed. Accordingly, the presence of the target in the search grid will be denoted $x \in \mathcal{A}$, and its absence, $x \notin \mathcal{A}$. The notation $x \in a$ and $x \notin a$ will analogously represent the presence or absence of the target in cell a.

The target search assumes the form of a decision making process in which the algorithm is tasked with determining the probability that the target is in the search area. This probability is assigned to a random variable H, such that

$$H = \begin{cases} 1 & \text{if } x_T \in \mathcal{A} \\ 0 & \text{if } x_T \notin \mathcal{A} \end{cases}$$

The $Pr(H = 1)$ is then called the aggregate belief probability, and is defined as

$$Pr(H = 1) = \sum_{a=1}^{|\mathcal{A}|} Pr(x_T = a)$$

The aggregate belief of the search area prior to the initialization of the search is defined as $Pr(H = 1) = \delta$.

A method for updating the aggregate belief probability through the search of the grid is then needed. The sensor for updating the belief will be modelled as a binary detector, in which its detections assume either a 0 or a 1, corresponding to either no target detection or target detection, respectively.

The model of the sensor will also account for the possibility of false alarms and missed detections, which we will call α and β, respectively. These parameters can be characterized for a given detector prior to use in a real search situation through either experimentation or specifications.

2.2.2 Bayesian Computation of Belief Probability

The Bayesian computation used in this discrete search is comprised of updating each individual cell probability after a measurement is taken. The basic equation for updating a cell, a is

$$Pr(x_T = a \,|D^t) = \frac{Pr(d_k^t \,|x_T = a, D^{t-1})}{Pr(d_k^t \,|D^{t-1})} Pr(x_T = a \,|D^{t-1})$$

where $Pr(x_T = a \,|D^t)$ is the belief probability that the target is in cell a at time t, $Pr(d_k^t \,|x_T = a, D^{t-1})$ is the probability of the detection d that you receive at time t given that the target is in cell a, $Pr(d_k^t \,|D^{t-1})$ is the total probability of detection d at time t, and $Pr(x_T = a \,|D^{t-1})$ is the prior belief of cell a. This formula is derived from Bayes' Law, which in its most trivial implementation assumes the form

$$P(A) \cdot P(B \,|A) = P(B) \cdot P(A \,|B)$$

The usefulness of using Bayesian probability to track your belief of the target's position comes in the fact that a program must only keep track of one number for each cell in the grid. In contrast with storing all of the measurements made over the course of the search, Bayes' rule allows us to use this update equation to fuse all of our previous information into one probability distribution, which is computationally efficient and useful in conserving computer memory on the small computers usually implemented on quadrotors. When a new measurement is taken, the update equation is used to compute the new distribution; in this respect, Bayes rule is memoryless, and therefore does not require the use of large amounts of computational power. For a more rigourous discussion of the derivation of this formula, we point to [1].

2.3 Algorithm Descriptions

The following search algorithms are described by Chung and Burdick [1] for use in discretized 2D search spaces. These algorithms will be used as a baseline to compare to more sophisticated strategies in continous search spaces in future research.

1. **Random Search:** The random search algorithm generates a random walk in which the agent will follow. This will occur at each time step, where the agent will randomly select a reachable cell in its vicinity. The random walk will serve as an upper bound time to detection for comparison with the other algorithms of interest. Note that the random search takes no advantage of the prior probability distribution.

2. **Sweeping Search:** The sweeping strategy follows the strategy of "sweeping" through the search grid, column by column. This algorithm is useful as another baseline comparison method, since it searches the grid in linear time. However, this technique also ignores the information provided by the prior probability distribution in designing its path. Since we will be testing the effects of obstacles in our simulations, this algorithm has been modified such that it uses an A^* search to produce a path to the next possible cell in its normal sweeping pattern.

3. **Optimal Lookahead Search:** The optimal lookahead search seeks to maximize some user defined utility function over the predetermined "lookahead window". The utility function, in this case, is the belief probability of target detection. The function uses the prior probability distribution to generate a path that will locally maximize the probability of detection. The "lookahead" window, w, describes the number of time steps over which the utility function will be examined. The possible trajectories over the next w steps are denoted $\pi = \{k_1, ..., k_w\}$. The optimal trajectory, π^*, is then described as

   $$\pi^* = \arg \max_{\pi=\{k_1,...,k_w\}} \sum_j^w Pr(x_T = k_j \left| D^t, d_{k_1}, ..., d_{k_{j-1}} \right) .$$

   Chung et al. provide a pseudocode of the algorithm in [1]. Note that when the lookahead window is set equal to one time step, the algorithm reduces to a greedy search (it is "greedy" because it moves to the neighboring cell with highest posterior probability). A^* will not be used in this search algorithm since this strategy implicitly plans its own path in contrast with several of the other algorithms, which only calculate a target cell and dont inherently plan the path to reach it.

4. **"Saccadic" Search:** The saccadic search derives its name from the phenomena exhibited by the human eye, in which it rapidly jumps from feature to feature. This search strategy allows the agent to effectively teleport to the cell with the highest prior belief probability, i.e.

   $$k^t = k_{max}^{t-1} \triangleq \arg\max_k (Pr(x_T = k \left| D^{t-1} \right))$$

   Although not physically realistic, it is interesting to note the similarities between this search and the Optimal Lookahead Search, as well as the "Drosophila-inspired" search, to be described next. This algorithm does not need to be modified in order to account for obstacles since it implicitly avoids them through saccading jumps.

5. **"*Drosophila*-inspired" Search:** This search is similar to the previously described "Saccadic" search in that it attempts to rapidly jump between peaks in the prior probability distribution however, this search executes a more physically feasible model. As opposed to "saccading" to probability peaks, this algorithm creates near-straight line paths between the current location and the maximum probability grid cell, i.e.

   $$k^t = \arg \min_{\Delta k < \kappa} (\|k_{max} - k^{t-1}\|)$$

   where Δk is the distance moved and κ is the maximum distance the agent can move in one time step. This algorithm offers a computionally cheap method of gaining information compared to other strategies like the Optimal Lookahead algorithm. The algorithm was inspired by the behavior of fruit flies (*Drosophila melanogaster*) which, in the pursuit of food, follow mostly straight line trajectories torward its current peak belief. Since this algorithm attempts to minimize the number of steps in order to reach the cell with the highest belief probability, A^* will also be used to here, in order to account for the possibility of obstacles lying what would otherwise be a straight line trajectory.

2.4 Experimental Description

An experiment has been proposed in order to examine the functionality of each of these search strategies in more realistic situations than has been previously studied. The simulations detailed by [1] do not include obstacles, are performed on a relatively small grid (ten by ten), and only describe the results for a truncated Gaussian distribution. The proposed experiment seeks to examine the effects of loosening these restrictions.

Again, the main parameters to be examined are the obstacle configuration and the prior probability distribution. A square grid will be used in the proposed simulation, with side lengths of 30 cells which should provide a sufficiently large search area, while keeping computational costs relatively low. The only downside to using a fixed grid size is that we are limiting ourselves to a fixed size, and therefore, the scalability of the results may be difficult to interpret.

Five different representative obstacle configurations will be used, the actual configuration of has yet to be decided. The proposed number of obstacle configurations has been limited to five in order to lower computational expenses of the simulation.

The prior probability distributions will be varied between uniform, Gaussian, Gaussian mixtures, and highly non-uniform. The covariances of the Gaussian distributions will assume four configurations (yet to be decided). The Gaussian mixtures will use different combinations of two, three, and four superposed Gaussians of fixed covariance, although the exploration of mixed Gaussians of different covariance may be an interesting topic to examine in the future. The highly non-uniform configurations may be created using different realizations, with certain cells having high probalities, but with surrounding cells have near-zero. The implementation of highly non-uniform may consist of assigning pseudo-random probabilities to cells, and choosing the real target location based on this random assignment. Another realization may generate random distributions that themselves are based on a family of Dirichlet prior distrbutions.

The variation in detector characteristics is another topic of study this experiment seeks to examine. High values for theα and β parameters (described above) may be provide unexpected results, since the model then assumes that the sensor is "lying" about the detections it observes. We propose that α and β between 0.2 and 0.8 in steps of 0.2 be examined. We will also study the effects of algorithm specific parameters, such as the lookahead window used in the Optimal Lookahead search, on the performance of the described algorithms.

In the current proposal, if all combinations of these variation of parameters were to be simulated, approximately 1000 different situations must be examined, times the number of trials that we wish to execute for each. For this reason, some decision making must be done in order to cut down on computational costs.

The variation of α and β, the different obstacle configurations, and the change of algorithm specific quantities like the lookahead window will be studied under a fixed mean and covariance Gaussian prior in the same sized search grid detailed above, and will be studied individually, as oppsed to varying all three characteristics simultaneously. Once the key parameters that determine search efficiency have been identified, the performance of each algorithm will be studied with regard to variation in the prior distribution. The inital experiment hopes to determine the pitfalls of each algorithm, such that it can then be optimized to handle varying prior distributions.

Another experiment in itself may consist of using these techniques with multiple searching agents, or even multiple targets, but we will leave such possibilities to future work.

References

[1] T.H. Chung and J.W. Burdick. "A Decision-Making Framework for Control Strategies in Probabilistic Search". In: *ICRA* (2007), pp. 4386–4393.