DISTANCE-PENALIZED ACTIVE LEARNING VIA MARKOV DECISION PROCESSES

Dingyu Wang and John Lipor

Department of Electrical & Computer Engineering
Portland State University
dingyu@pdx.edu, lipor@pdx.edu

Gautam Dasarathy

School of Electrical, Computer,
& Energy Engineering
Arizona State University
gautamd@asu.edu

ABSTRACT

We consider the problem of active learning in the context of spatial sampling, where the measurements are obtained by a mobile sampling unit. The goal is to localize the change point of a one-dimensional threshold classifier while minimizing the total sampling time, a function of both the cost of sampling and the distance traveled. In this paper, we present a general framework for active learning by modeling the search problem as a Markov decision process. Using this framework, we present time-optimal algorithms for the spatial sampling problem when there is a uniform prior on the change point, a known non-uniform prior on the change point, and a need to return to the origin for intermittent battery recharging. We demonstrate through simulations that our proposed algorithms significantly outperform existing methods while maintaining a low computational cost.

***Index Terms*—** Active learning, adaptive sampling, autonomous systems, mobile sensor, path planning.

1. INTRODUCTION

In this paper, we consider active learning [1] in the context of spatial sampling, where measurements are obtained by a mobile sampling unit. A natural subproblem in this setting is one of localization of a change-point of a one dimensional threshold classifier. Such problems have been cast and solved using recursive bisection (a la *binary search*) in wide range of problems in computer science and engineering. For instance, bisection procedures have been featured in classification [2, 3], clustering [4], graph annotation [5], and multi-armed bandit problems [6]. In all cases, the assumption is that there is a fixed, though perhaps non-uniform, cost of obtaining samples, and the goal is to minimize the total cost incurred while performing inference with a specified degree of certainty. However, in the context of environmental sensing and monitoring, obtaining such samples typically corresponds to traveling from location to location, and hence the sampling cost is a function of both the number of measurements taken *and* the distance traveled while sampling.

This work is motivated by the above situation, also considered in [7–9], in which we wish to determine the spatial extent/boundary of a phenomenon of interest using as little *time* as possible. The authors of [8] show that a two-dimensional search problem can be broken into a series of one-dimensional searches, each of which amounts to finding the threshold of a step function on the unit interval (see Fig. 1). They then present a search algorithm called Quantile Search (QS) that explicitly trades off between the number of samples taken and the distance traveled during the search procedure. While QS outperforms traditional binary search, as well as greedy algorithms such as that presented in [10], the question remains of whether this method is optimal in the sense of achieving the lowest possible sampling time.

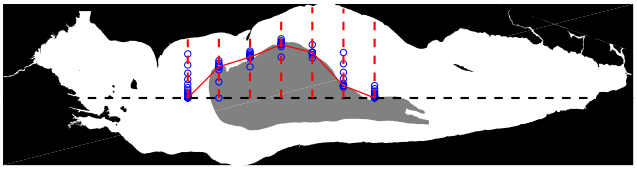


Fig. 1. Example boundary estimation problem. The two-dimensional boundary can be estimated via a series of one-dimensional searches along dashed vertical lines [8].

In this paper, we present a general framework for distance-penalized active learning by drawing novel connections between active learning and Markov decision processes (MDPs). By viewing the search procedure as a sequential hypothesis/version space reduction [11], we re-cast the active learning problem as a stochastic shortest path (SSP) problem [12, Ch. 2], where the states correspond to all possible lengths of the hypothesis space. This novel connection allows us to obtain an optimal solution to the one-dimensional search problem that significantly outperforms existing methods in terms of expected time spent during the search procedure. We then show how to incorporate more general costs into this framework, including non-uniform priors on the change point and the need to return to the origin for intermittent battery recharging.

2. PROBLEM FORMULATION & RELATED WORK

Following [2, 9], we focus on the one-dimensional search problem and note that the two-dimensional spatial sampling problem can be reduced to a series of one-dimensional searches, as displayed in Fig. 1. Assume the underlying function for some quantity is a member of the step function class $\mathcal{F}$, defined as

$$\mathcal{F} = \{f : [0,1] \to \mathbb{R} \mid f(x) = \mathbb{1}_{[0,\theta)}(x)\},$$

where $\theta \in [0,1]$ is the *change point* and $\mathbb{1}_S(x)$ denotes the indicator function which is 1 on the set S and 0 elsewhere. Given a fixed unknown θ, the search begins at the initial location $X_0 = 0$, proceeds by obtaining binary-valued measurements $Y_n = f(X_n)$, and terminates when the change point θ is known to lie within an interval of length at most ε.

The goal is to complete the search while minimizing the total time required for sampling, a function of both the number of samples taken and the distance traveled by the vehicle carrying out the search

task. Specifically, given the number of samples N and the locations of each sample $\{X_n\}_{n=1}^N$, the total time is

$$T_{\text{tot}} = T_s N + T_t \left(\sum_{n=1}^{N} |X_n - X_{n-1}| \right), \tag{1}$$

where T_s is the time required to obtain a single sample and T_t is the time required to travel a unit distance. To find an optimal search strategy, one may wish to minimize (1) in expectation with respect to some prior distribution on θ.

We also view the search problem as a binary classification problem, in which the feature space is $\mathcal{X} = [0, 1]$, the label space is $\mathcal{Y} = \{0, 1\}$, and we wish to select a hypothesis h corresponding to the change point θ from the *hypothesis space* $\mathcal{H} = [0, 1]$. Each measurement removes a number of hypotheses from $\mathcal{H}$, yielding an updated hypothesis space $\mathcal{H}_n$ of size less than that of $\mathcal{H}$. In this light, the search procedure terminates when the length of the updated hypothesis space $|\mathcal{H}_N|$ has been reduced to be at most ε.

2.1. Related Work

Algorithms designed to search for the change point of a step function typically focus on minimizing only the number of samples taken throughout the procedure, resulting in a binary search or bisection-type solution [2, 13–18]. These methods focus on the noisy model, in which the measurements Y_n are flipped with some known probability. In [13] the authors propose the probabilistic binary search algorithm, which maintains a Bayesian estimate of the posterior distribution on θ given all previous measurements and proceeds by obtaining measurements that bisect this posterior. This algorithm was analyzed in [14], and its optimality properties are studied in [2, 17]. While these results have significant applications in a number of domains, their myopic nature inhibits the inclusion of general costs, such as the distance traveled throughout the search procedure or the need to return to a base station to recharge a vehicle.

The problem of active learning for spatial sampling is also considered in [19–21], where the authors assume the underlying potential function is a Gaussian process. However, these approaches require the selection of an appropriate kernel, as well as a number of hyperparameters, both of which require existing data to fit. Further, in this work we consider the case of binary-valued measurements, yielding algorithms that are optimal in a deterministic sense, as opposed to providing high-probability regret bounds.

Most relevant to our presented work is the quantile search algorithm presented in [7, 8]. The key idea behind the QS algorithm is that a trade-off between the number of samples taken and the distance traveled in a search procedure can be obtained by sampling at points $1/m$ into the hypothesis space, where $m \geq 2$ is a tuning parameter. When $m = 2$ the algorithm reduces to binary search, and as $m \to \infty$ the algorithm tends toward continuous sampling. The authors also present a variant of QS that naturally extends the work of [2, 14] to allow for noisy measurements. More recently, [9] presents an extension of the QS algorithm, called Uniform-to-Binary (UTB) search, that decreases the search parameter m as the hypothesis space is reduced. The intuition behind this method is that as the hypothesis space shrinks, the algorithm can be more aggressive in terms of information gain, since the maximum possible overshoot of the change point θ is smaller. While UTB improves on standard QS significantly, the question remains of whether it is the optimal solution to the proposed search problem. In this work, we provide such an optimal solution and show that the resulting policy obtained by the MDP framework outperforms QS and UTB.

MDPs constitute a well-studied model for sequential decision making with broad applications in controls, scheduling, and robotics [12, 22]. While MDPs and dynamic programming have been applied to robotic search problems in the past, these focus only on the case where samples can be obtained continuously (see [20, 23] and the references therein). The type of MDP we are interested in is the stochastic shortest path (SSP) model [12, Ch. 2], where the goal is to reach a terminal state at minimal cost. The SSP formulation is traditionally applied to robotic planning problems where the states correspond to the set of physical locations the vehicle can visit and the terminal states correspond to known locations of end goals [24, 25]. In contrast, we model the terminal states as those in which the hypothesis space is sufficiently reduced, connecting the SSP problem with a common approach to active learning.

3. SHORTEST-PATH SEARCH

In this section, we define the stochastic shortest path problem and describe the key insight behind our proposed approach, which we refer to as *Shortest-Path Search* (SPS). A SSP problem is defined by a 5-tuple (S, A, P, C, T), where S is a set of states taken by the system of interest, A is a set of actions, $P(s'|s, a)$ is the state transition kernel equal to the probability of transitioning from state s to state s' after taking action a, $C(s, a)$ is is the real-valued cost of taking action a at state s, and finally T is the set of terminal states having zero cost. A policy is a function $\pi : S \to A$ that maps every state to an action, and the goal is to determine a policy that minimizes the expected cost to reach a terminal state, i.e., to solve

$$\min_{\pi} \mathbb{E}\left[\sum_{n=1}^{N} C\left(s_n, \pi(s_n)\right) \right] \tag{2}$$

where $s_n \in S$ and it is assumed that $s_{N+k} \in T$ for all $k \in \mathbb{N}$.

Our key observation is that existing search strategies such as binary search and QS maintain a single interval as the hypothesis space and are completely defined by how far they travel into the current hypothesis space $\mathcal{H}_n$. For example, binary search always samples half way into $\mathcal{H}_n$, whereas QS samples $1/m$ into $\mathcal{H}_n$ for some $m \geq 2$. Formally, this class of search strategies may be indexed by the set of functions $\pi : [0, 1] \to [0, 1]$ that map from $|\mathcal{H}_n|$ to the proportion of the hypothesis space the algorithm travels. We consider this class of algorithms, for which we may obtain optimal solutions to (2). With this insight in mind, we define the state space S as the set of possible lengths of the hypothesis space and the action space A as the set of proportions to travel into the current hypothesis space. To obtain a finite number of possible states and actions, we discretize the unit interval into bins of length $\Delta \in \mathbb{R}$, which is also done in [2, 8, 14]. The state transition probability then denotes the probability (with respect to the randomness in θ) of achieving a hypothesis space of length s' after taking action a while in state s. For example, QS with $m = 3$ begins with $s_0 = 1$ and performs the action $a_1 = 1/3$, yielding $\mathbb{P}(s_1 = 1/3) = 1/3$ and $\mathbb{P}(s_1 = 2/3) = 2/3$ when we assume a uniform prior on θ. The resulting cost is the time to acquire a single sample plus the time to travel the distance defined by the action chosen, and hence (2) becomes

$$\min_{\pi} \mathbb{E}_{\theta}\left[\sum_{n=1}^{N} T_s + T_t \left(|\mathcal{H}_n| \, \pi\left(|\mathcal{H}_n| \right) \right) \right], \tag{3}$$

which minimizes the quantity defined in (1).

The optimal solution to (3) may be obtained by performing *value iteration*, which is known to yield the optimal policy for the SSP

Algorithm 1 Value Iteration for Shortest-Path Search

Input: 5-tuple (S, A, P, C, T)
Output: Optimal search policy π^*, value function V^*
Initialize: $\pi^*, V^* \in \mathbb{R}^{|S|}$ arbitrary
for $s \in T$ **do**
    $V^*(s) \leftarrow 0$
end for
Sort S by size of hypothesis space
for $s \in S$ (traverse in sorted order) **do**
    $B(a) \leftarrow C(s,a) + \sum_{s' \prec s} P(s' \mid s,a) V^*(s')$ for all $a \in A$
    $\pi^*(s) \leftarrow \arg\min_{a \in A} B(a)$
    $V^*(s) \leftarrow \min_{a \in A} B(a)$
end for

problem [12, Ch. 2]. However, the computational complexity of value iteration may be significantly reduced by noting that we may assign a total order on the states via the length of the hypothesis space. The resulting state transition graph cannot be cyclic, since any meaningful sampling procedure will decrease the size of the hypothesis space. Thus in order to calculate the optimal policy from state s, it is enough to know the values of the set of "smaller" states $\{s' \mid |\mathcal{H}_{s'}| < |\mathcal{H}_s|\}$ [26]. Sorting the states according to this order significantly reduces the computational complexity of value iteration. We include pseudocode for this procedure in Algorithm 1.

We now describe the tuple (S, A, P, C, T) for a variety of practical scenarios. We demonstrate how our framework allows for the seamless incorporation of prior knowledge on the distribution of the change point θ, as well as the more sophisticated scenario where the vehicle needs to be recharged after a fixed period of time. We stress that the resulting policies are *optimal* in the sense of minimizing the total sampling time (1) and do not require the number of samples to be known a priori. In all cases, we discretize the unit interval into $\Delta^{-1} \in \mathbb{N}$ bins, each having length Δ, and fix a minimum estimation error ε such that $\Delta < \varepsilon$.

3.1. Uniform Prior on Change Point

Absent prior information on the change point θ, it is natural to begin with the uniform distribution on the unit interval. As stated in Section 3, the states of our proposed MDP are the set of possible lengths of the hypothesis space, and the actions are the proportion of the hypothesis space to travel. Therefore, we take $S = \{\Delta, 2\Delta, \ldots, 1-\Delta, 1\}$ and $A = \{\Delta, 2\Delta, \ldots, 1-\Delta\}$, noting that while in state $s = i\Delta$, the valid actions $a = k\Delta$ are only those with $k \leq i$. Each sample/action places the system in one of two possible states, depending on whether the change point is to the left or the right of the sample location. Hence, given a uniform prior on θ, the resulting state transition probability is

$$P(s' \mid s = i\Delta, a = k\Delta) = \begin{cases} \frac{k}{i}, & s' = k\Delta \\ \frac{i-k}{i}, & s' = (i-k)\Delta \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

We set the cost of each action to be $C(s, k\Delta) = T_s + T_t(k\Delta)$, and terminal states are those with $i\Delta \leq \varepsilon$. With the intelligent computation described in Alg. 1, the optimal policy may be computed in $O(\Delta^{-2})$ time.

3.2. Non-Uniform Prior on Change Point

Next, we consider the case where we have prior knowledge about the location of θ that is encoded as a prior distribution. Such a setting may appear in practical applications when the goal is to constantly monitor the spatial extent of some phenomenon. In this case, the estimated boundary in the previous cycle serves as a prior for the current boundary.

Encoding the search as an MDP is more difficult in this case, since there may exist multiple states with the same hypothesis space length but different priors on θ. However, the following proposition shows that even in this setting, the current prior maybe updated purely using the current hypothesis space. Therefore, the states can be encoded entirely by the two end points of the current hypothesis space, augmenting the definition from the previous section.

Proposition 1. *Assume we are given a prior p on the location of the change point θ and that a search procedure has collected n arbitrary samples, resulting in the current hypothesis space $\mathcal{H}_n$. Then the updated prior on θ is*

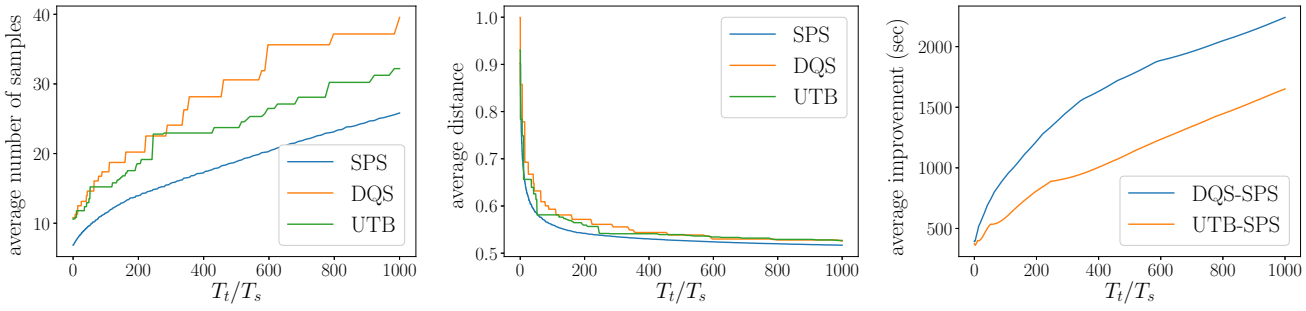
$$p_n(\theta) = \begin{cases} \frac{p(\theta)}{\int_{\mathcal{H}_n} p(\tau) \mathrm{d}\tau}, & \theta \in \mathcal{H}_n \\ 0, & \textit{otherwise} \end{cases}. \quad (5)$$

The proof proceeds via Bayes' rule and is omitted due to spatial constraints. Given an initial prior p on θ, Prop. 1 allows us to write the state transition probabilities in closed form, again computing the optimal policy using value iteration. The corresponding state space must be augmented to account for both end points of the hypothesis space, i.e., $S = \left\{(i\Delta, j\Delta) : i, j \in \left\{0, 1, \ldots, \Delta^{-1}\right\}\right\}$, increasing the computational complexity to $O(\Delta^{-3})$. The action space and cost remain unchanged, and the set of terminal states is $T = \{(i\Delta, j\Delta) \in S : |i-j|\,\Delta \leq \varepsilon\}$.

3.3. Intermittent Recharging

In this section, we describe how our proposed MDP-based formulation can be used to incorporate the need for the vehicle to return to the origin for intermittent battery recharging. Such a scenario may occur frequently in practice when deploying the proposed algorithm on unmanned aerial vehicles (drones), which have a battery life of around thirty minutes [27]. In this case, the policy should take into account both the hypothesis space *and* the remaining battery life. This can be accounted for by further state space augmentation, albeit at the cost of higher computational complexity. In particular, assume the vehicle has a lifetime of T_0 when fully charged, and that we discretize time into units of size δ. We then let the state space be $S = \left\{(i\Delta, j\Delta, t\delta) : i, j \in \left\{0, 1, \ldots, \Delta^{-1}\right\}, t \in \left\{0, 1, \ldots, T_0\delta^{-1}\right\}\right\}$, where $t\delta$ denotes the remaining battery life of the vehicle. The action space is also augmented to include the recharge time, and hence actions are of the form $a = (k\Delta, l\delta)$, where $k\Delta$ denotes the portion of the hypothesis space to travel and $l\delta$ denotes the time to spend charging. Under the uniform prior assumption, the transition probabilities are straightforward to compute. The cost is defined as $C(s, (k\Delta, l\delta)) = T_s + T_t(k\Delta) + l\delta$, where we note that $l = 0$ in the case where recharging is not performed. The terminal states are $T = \{(i\Delta, j\Delta, t\delta) : |i-j|\,\Delta \leq \varepsilon\}$.

The optimal policy can be computed using a modified form of Alg. 1, and we note that in this setting a number of states and actions may be ignored during value iteration. For example, the vehicle will never enter a state in which it does not have enough battery to return to the origin, nor will it attempt to sample at a location if the combined travel plus sampling time exceeds the maximum battery life. Similarly, actions that would result in either of these situations may be ignored. A full description of allowable states and actions, as well as detailed transition probabilities for this and the previous section, are deferred to a later report due to spatial constraints.



(a) (b) (c)

Fig. 2. Performance as a function of T_t/T_s. (a) Average number of samples. (b) Average distance traveled. (c) Improvement in average total sampling time between QS/UTB and proposed SPS algorithm.

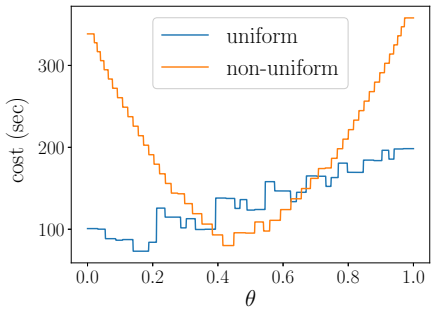


Fig. 3. Total sampling time as a function of change point θ for SPS algorithm with uniform prior (mean cost 130.56 sec) and non-uniform prior (mean cost 105.61 sec).

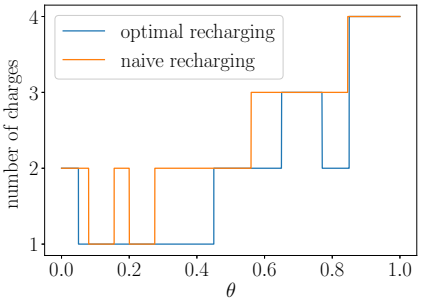


Fig. 4. Number of charges as a function of change point θ for SPS algorithm with optimal recharging (mean 2.02 charges) vs naive recharging (mean 2.45 charges).

4. SIMULATIONS

In this section, we implement the models constructed in Section 3 and compare our proposed SPS policy with QS and UTB. We first consider the case where θ is distributed uniformly on the unit interval. We compare the performance of the three algorithms for a variety of ratios of travel cost T_t to sample cost T_s, noting that the policy that minimizes (1) depends only on this ratio (aside from Δ and ε). We discretize the unit interval into bins of size $\Delta = 0.001$ and terminate when we reach a state of size at most $\varepsilon = 0.01$. We fix $T_s = 100$ and let T_t range over a uniform grid of 1000 values in the range 0.0001-1000. For each value of T_t, we choose the best m for QS and UTB separately and simulate the expected cost for θ ranging over a uniform grid of 1000 points. Fig. 2 shows the resulting number of samples, distance traveled, and difference in total cost between QS/UTB and the SPS algorithm as a function of the ratio T_t/T_s. Interestingly, the SPS algorithm typically uses far fewer samples than either QS or UTB while traveling approximately the same distance (on average). This is manifested in Fig. 2(c), where we see that both QS and UTB have a total sampling time up to 30 minutes greater than the optimal algorithm.

We next evaluate the performance of SPS when there is a non-uniform prior on θ. We set $T_s = 10, T_t = 100, \Delta = 0.001, \varepsilon = 0.01$ and let the prior p be the truncated normal distribution (with support [0,1]) having mean 0.5 and variance 0.1. With these settings, we compute the optimal non-uniform SPS policy, as described in Section 3.2. The resulting cost (in seconds) as a function of θ shown in Fig. 3. The figure shows a clear trade-off made by incorporating non-uniform priors—the cost is higher in the low-frequency regions (away from 0.5, as defined by the prior), but the cost is much lower in high-frequency regions. This results in a significantly lower average sampling time for data drawn from this distribution. Averaging over 1000 random instances of θ from the distribution p, the optimal uniform-prior policy results in an average sampling time of 130.56 sec compared to 105.51 sec when incorporating a non-uniform prior.

Finally, we demonstrate the efficacy of incorporating battery recharging into the search problem. We set $T_s = 40, T_t = 100, T_0 = 250, \Delta = 0.01, \delta = 1$, and $\varepsilon = 0.04$, taking a uniform prior on the change point, and run 10,000 instances of the optimal algorithm with θ drawn according to the uniform distribution. Fig. 4 shows the number of charges required as a function of the change point for both the optimal policy described in Section 3.3 and a naive policy that recharges myopically. Incorporating recharging into the policy results in an average of 17% fewer returns to the origin for battery recharging. For change points in the vicinity of 0.4, this results in a dramatic time savings, as well as improved efficiency of vehicle usage in terms of battery utilization.

5. CONCLUSION

We have presented a general framework for active learning where the cost is a function of the number of measurements taken *and* the distance traveled by the sampling vehicle. The resulting algorithms are derived by casting the one-dimensional search problem as a stochastic shortest path MDP, for which optimal solutions are obtained using value iteration. The proposed SPS algorithm significantly outperforms existing methods in terms of total sampling cost.

6. REFERENCES

[1] Burr Settles, *Active Learning*, Morgan & Claypool, 2012.

[2] Rui Castro and Robert Nowak, "Minimax bounds for active learning," *IEEE Trans. Inf. Theory*, vol. 54, pp. 2339–2353, May 2008.

[3] Robert Nowak, "The geometry of generalized binary search," *IEEE Trans. Inf. Theory*, vol. 57, pp. 7893–7906, 2011.

[4] Hassan Ashtiani, Shrinu Kushagra, and Shai Ben-David, "Clustering with same-cluster queries," in *Advances in neural information processing systems*, 2016, pp. 3216–3224.

[5] Gautam Dasarathy, Robert Nowak, and Xiaojin Zhu, "S2: An efficient graph based active learning algorithm with application to nonparametric classification," in *Conference on Learning Theory*, 2015, pp. 503–522.

[6] Zohar Karnin, Tomer Koren, and Oren Somekh, "Almost optimal exploration in multi-armed bandits," in *International Conference on Machine Learning*, 2013, pp. 1238–1246.

[7] John Lipor and Laura Balzano, "Quantile search: A distance-penalized active learning algorithm for spatial sampling," in *Proc. Allerton Conf. on Communication, Control, and Computing*, 2015.

[8] John Lipor, Brandon P Wong, Donald Scavia, Branko Kerkez, and Laura Balzano, "Distance-penalized active learning using quantile search," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5453–5465, 2017.

[9] John Lipor and Gautam Dasarathy, "Quantile search with time-varying search parameter," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1016–1018.

[10] Pinar Donmez and Jaime G Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 619–628.

[11] Robert Nowak, "Generalized binary search," in *Proc. Allerton Conference on Communication, Control, and Computing*, 2008.

[12] Dimitri P. Bertsekas, *Dynamic programming and optimal control*, vol. 2, Athena scientific Belmont, MA, 2007.

[13] M. Horstein, "Sequential decoding using noiseless feedback," *IEEE Trans. Inf. Theory*, vol. 9, 1963.

[14] M. V. Burnashev and K. S. Zigangirov, "An interval estimation problem for controlled observations," *Problems in Information Transmission*, vol. 10:223-231, 1974, Translated from Problemy Peredachi Informatsii, 10(3):51-61, July-September, 1974.

[15] Richard M. Karp and Robert Kleinberg, "Noisy binary search and its applications," in *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[16] Michal Ben Or and Avinatan Hassidim, "The bayesian learner is optimal for noisy binary search (and pretty good for quantum as well)," in *Proc. IEEE Symposium of Foundations of Computer Science*, 2008.

[17] Bruno Jedynak, Peter I Frazier, and Raphael Sznitman, "Twenty questions with noise: Bayes optimal policies for entropy loss," *Journal of Applied Probability*, vol. 49, no. 1, pp. 114–136, 2012.

[18] Rolf Waeber, Peter I. Frazier, and Shane G. Henderson, "Bisection search with noisy responses," *SIAM Journal on Control and Optimization*, vol. 51, pp. 2261–2279, 2013.

[19] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause, "Active learning for level set estimation," in *IJCAI*, 2013, pp. 1344–1350.

[20] Gregory Hitz, Alkis Gotovos, Marie-Éve Garneau, Cédric Pradalier, Andreas Krause, Roland Y Siegwart, et al., "Fully autonomous focused exploration for robotic environmental monitoring," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2658–2664.

[21] Ilija Bogunovic, Jonathan Scarlett, Andreas Krause, and Volkan Cevher, "Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation," in *Advances in Neural Information Processing Systems*, 2016, pp. 1507–1515.

[22] Dimitri P. Bertsekas, *Dynamic programming and optimal control*, vol. 1, Athena scientific Belmont, MA, 2005.

[23] Geoffrey A. Hollinger, Sunav Choudhary, Parastoo Qarabaqi, Christopher Murphy, Urbashi Mitra, Guarav S. Sukhatme, Milica Sotjanovic, Hanumant Singh, and Franz Hover, "Underwater data collection using robotic sensor networks," *IEEE Journal on Sel. Areas in Comm.*, vol. 30, pp. 899–911, 2012.

[24] Levente Kocsis and Csaba Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*. Springer, 2006, pp. 282–293.

[25] Felipe W Trevizan and Manuela M Veloso, "Short-sighted stochastic shortest path problems.," in *ICAPS*, 2012.

[26] Peng Dai, Daniel S Weld, Judy Goldsmith, et al., "Topological value iteration algorithms," *Journal of Artificial Intelligence Research*, vol. 42, pp. 181–209, 2011.

[27] DJI, "Matrice 100: The quadcopter for developers," https://www.dji.com/matrice100.