A Bayesian Active Learning Approach to Adaptive Motion Planning

Sanjiban Choudhury and Siddhartha S. Srinivasa

Abstract An important requirement for a robot to operate reliably in the real world is a robust motion planning module. Due to limited on-board sensing and computation, state of the art motion planning systems do not have consistent performance across all situations a robot encounters. We are interested in planning algorithms that *adapt during a planning cycle* by actively inferring the structure of the valid configuration space, and focusing on potentially good solutions.
Consider the problem of evaluating edges on a graph to discover a good path. Edges are not alike in value - some are *important*, others are *informative*. Important edges have a lot of good paths flowing through them. Informative edges, on being evaluated, affect the likelihood of other neighboring edges being valid. Evaluating edges is expensive, both for robots with complex geometries like robot arms, and for robots with limited onboard computation like UAVs. Until now, we have addressed this challenge via *laziness*, deferring edge evaluation until absolutely necessary, with the hope that edges turn out to be valid. Our key insight is that we can do more than passive laziness - we can *actively* probe for information. We draw a novel connection between motion planning and Bayesian active learning. By leveraging existing active learning algorithms, we derive efficient edge evaluation policies which we apply on a spectrum of real world problems. We discuss insights from these preliminary results and potential research questions whose study may prove fruitful for both disciplines.

1 Introduction

Motion planning, the task of computing collision-free motions for a robotic system from a start to a goal configuration, has a rich and varied history [17]. Up until now, the bulk of the prominent research has focused on the development of tractable planning algorithms with provable *worst-case performance guarantees* such as computational complexity [3], probabilistic completeness [18] or asymptotic optimality

Sanjiban Choudhury[1] and Siddhartha S. Srinivasa[2]
[1]The Robotics Institute, Carnegie Mellon University e-mail: sanjiban@cmu.edu, [2]School of Computer Science and Engineering, University of Washington e-mail: siddh@cs.uw.edu

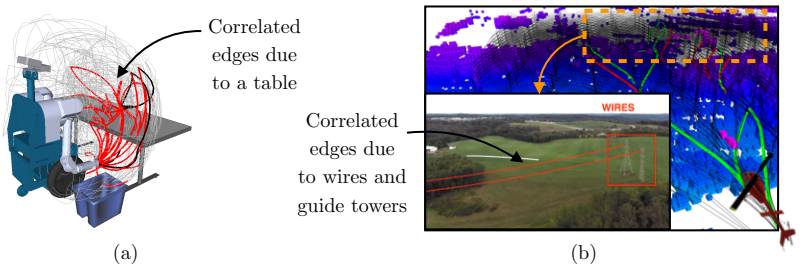

Fig. 1 Real world planning problems where edges are correlated. In such cases, a *white-box adaptive* planner can infer the structure of the world from outcomes of edge evaluations. (a) Presence of a table in robotic arm planning correlates neighbouring edges (courtesy Dellin [9]). (b) Presence of wires and guide-towers in helicopter planning correlates corresponding edges.

[16]. In contrast, analysis of the *expected performance* of these algorithms on the real world planning problems a robot encounters has received considerably less attention, primarily due to the lack of standardized datasets or robotic platforms. However, recent advances in affordable sensors and actuators have enabled mass deployment of robots that navigate, interact and collect real data. This motivates us to examine new algorithmic questions such as: "How can we design planning algorithms that, subject to on-board computation constraints, maximize their expected performance on the actual distribution of problems that a robot encounters?"

We formalize the problem as follows - given a probability distribution over worlds, we want to optimize expected planning effort of finding a good solution. The main planning effort is in collision checking as it requires expensive geometric intersection computations [1, 20, 20, 10, 9]. Moreover, the uncertainty over worlds induces a complex uncertainty over the validity of edges. To solve such problems, we advocated in [24] for an *adaptive motion planning system* that learns which potential solutions to evaluate using features extracted from the world. However, a significant drawback is that feature extraction requires extensively probing the environment before any planning can commence. Moreover, the planners are treated as a "black box" wherein no degree of intervention during the planning cycle is allowed. This motivates us to examine the "white box" paradigm where a planner *adapts during planning*. Often a large degree of inference can be made about the world in the intermediate stages of search as shown in Fig. 1. Based on this information, the planning algorithm itself can choose to adapt its search strategy and make the most out of its planning effort.

While the white box paradigm encompasses a large class of problems, in this paper we focus on one such subclass - adaptive edge evaluation during search on explicit graphs. Consider a graph whose vertices represent robot configurations and edges represent potentially valid movements between these configurations. In order to minimize edge evaluation effort, prior information about the validity of edges can be leveraged [6, 2]. This is illustrated in Fig. 2 where a robot navigates in a world with a narrow gap in one of two places. If an edge on gap 1 is blocked, the planner infers that gap 2 is free. In other words, the posterior distribution over worlds collapses on a set of worlds, all of which admit a particular path to be feasible. The planner

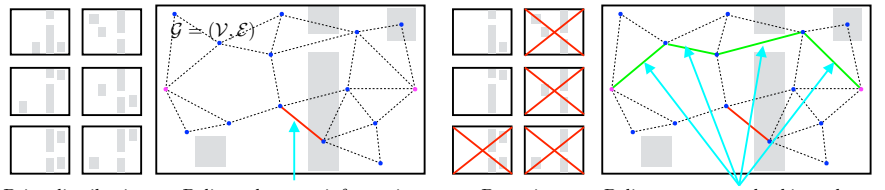

Fig. 2 Collision checking informative edges can lead to reduced search effort. (a) A prior distribution over worlds shows a narrow gap in one of two places. Hence the collision checker checks the most informative edge and finds it to be in collision. (b) The posterior collapses on one path being free (but many possible worlds). The algorithm proceeds to check these edges for collision.

proceeds to check only that path, leaving all other edges untouched. We wish to compute such a policy that judiciously chooses edges to evaluate by reasoning about likely worlds in which the robot operates.

Our key insight is that this problem is equivalent to the Bayesian active learning problem of *decision region determination (DRD)* [15, 4] - given a set of tests (edges), hypotheses (worlds), and regions (potential paths), the objective is to select a sequence of tests that drive uncertainty into a single decision region. Drawing this analogy enables us to leverage existing methods in Bayesian active learning [12] for robot motion planning. Note that the DRD problem has one key distinction from the general active learning problem - the uncertainty only needs to be driven down enough to ascertain if a path is free, and it is not necessary to identify the underlying world [8]. This makes it very applicable for motion planning where we only need to know enough about the world to compute a path. Solving the DRD problem in general is NP-hard [15]. Fortunately, Chen et al.[4] provide a method to solve this problem by maximizing an objective function that satisfies *adaptive submodularity* [11] - a natural diminishing returns property that endows greedy policies with near-optimality guarantees. We are able to directly adopt this algorithm to compute a policy that would select which edges to evaluate for our problem.

We are excited to make this connection between two disparate disciplines. We have made some preliminary inroads and demonstrated great empirical results compared to several state of the art baseline algorithms in real world settings [5]. Interestingly, we found that this connection leads to new subproblems that necessitate attention from the active learning community. One such problem, as discussed in Section 3, is the DRD problem when test outcomes are independent Bernoulli random variables. We show that under such a setting the computational complexity for edge evaluation is reduced from exponential in number of edges to linear while still retaining near-optimality guarantees. We describe other such problem variants, insights and potential future research directions in detail in Section 4.

2 Problem Formulation

We now describe the adaptive edge evaluation, drawing the equivalence with the DRD problem along the way. Let $G = (V, E)$ be an explicit graph that consists of a set of vertices V and edges E. Given a pair of start and goal vertices, $(v_s, v_g) \in V$, a search algorithm computes a path $\xi \subseteq E$ - a connected sequence of valid edges.

We have a set of worlds $H = \{h_1, \ldots, h_n\}$, each of which are analogous to a "hypothesis". We have a prior distribution $P(h)$ on this set. A "test" is performed by querying an edge $e \in E$ for evaluation which returns a binary outcome $x \in \{0, 1\}$ denoting if an edge is valid or not. Thus each world $h \in H$ can be considered a function $h : E \to \{0, 1\}$ mapping edges to corresponding outcomes. We address applications where edge evaluation is expensive, i.e., the computational cost $c(e)$ of computing $h(e)$ is significantly higher than regular search operations.

We make a second simplification to the problem - from that of search to that of identification. Instead of searching G online for a path, we frame the problem as identifying a valid path from a library of 'good' candidate paths $\Xi = (\xi_1, \xi_2, \ldots, \xi_m)$. Each path can be thought of as carving out a "decision region" over the space of worlds. By abuse of notation, we write $\xi \subseteq H$ to denote that each path corresponds to a set of worlds for which it would be feasible.

If a set of edge evaluations $S \subseteq E$ are performed, let the observed outcome vector be denoted by $\mathbf{x}_S$. Let the version space $H(\mathbf{x}_S)$ be the set of worlds consistent with observation vector $\mathbf{x}_S$, i.e. $H(\mathbf{x}_S) = \{h \in H \mid \forall e \in S, h(e) = \mathbf{x}_S(h)\}$.

We define a policy π as a mapping from observation vector $\mathbf{x}_S$ to edges. A policy terminates when it shows that at least one path is valid, or all paths are invalid. Let h be the underlying world on which it is evaluated. Denote the observation vector of a policy π as $\mathbf{x}_S(\pi, h)$. The expected cost of a policy π is $c(\pi) = \mathbb{E}_h[c(\mathbf{x}_S(\pi, h)]$ where $c(\mathbf{x}_S)$ is the cost of all edge evaluations $e \in S$. The objective is to compute a policy π^* with minimum cost that ensures at least one path is valid, i.e.

$$\pi^* \in \arg\min_{\pi} c(\pi) \text{ s.t } \forall h, \exists \xi_d \ : \ P(\xi_d \mid \mathbf{x}_S(\pi, h)) = 1 \tag{1}$$

3 Approach

We adopt the framework of *Decision Region Edge Cutting (*DIRECT*)* [4] which we describe briefly in our context. DIRECT creates a graph where nodes are hypotheses (worlds) and edges are between hypotheses belonging to different regions. Performing a test 'cuts' edges if any one of the hypotheses is inconsistent with the test outcome. Once all the edges are cut, the uncertainty collapses into atleast one of the regions. A key difficulty arises from the fact that regions overlap. DIRECT solves this problem

by defining a set of subproblems containing disjoint regions with the caveat that solving any one subproblem suffices, and combining subproblems with a Noisy-OR operator. Interstingly, DIRECT can directly solve our edge evaluation problem!

A concern is that DIRECT requires $|H|$ computation per sub-problem, which can be $\mathcal{O}\left(2^E\right)$. We circumvent this problem by computing a decision tree offline using DIRECT which has a runtime of $\mathcal{O}(1)$. The nodes of the tree encode which edge to evaluate. The tree branches on the outcome of the evaluation. The tree terminates on a leaf node when the uncertainty has been pushed onto one region. The tree also terminates if there are no consistent worlds in its database that matches the outcome vector.

Hence, if the world at test time is not in the training data, it may lead to a situation where the offline decision tree terminates without finding a solution. We need a policy to execute online under such situations. However, we would still like the policy to be informed by some prior. Since assuming independent edges is a common simplification [17, 19, 6, 2, 9], we assume edges are independent Bernoulli random variables. This leads to a new active learning problem definition. While naively applying DIRECT requires $\mathcal{O}\left(2^E\right)$ per iteration, we present a more efficient *Bernoulli Subregion Edge Cutting (*BISECT*)* algorithm [5], which computes each subproblem in $\mathcal{O}(E)$ time.

4 Discussion and Future Directions

We empirically evaluated BISECT on a spectrum of synthetic and real world planning problems [5]. These results demonstrate the efficacy of leveraging prior data to significantly reduce collision checking effort. Interestingly, they raise a lot of research questions which we discuss below.

Q 1. *We need to relax assumptions in the framework in (2) - the prior is specified only via a finite database of worlds and selection is limited to a fixed library of paths. (a) Can we better model how collision information propagates through the graph? (b) Can we circumvent explicitly computing a candidate set of paths?*

(a) Specifying the prior as a finite database of worlds is memory inefficient and can lead to overfitting. A better way is to build and update belief distributions over configuration space using techniques such as KDE [6], mixture of Gaussians [14], RKHS [23] or even customized models [22]. The efficacy of these models depends on how accurately they can represent the world, how efficiently they can be updated and how efficiently they can be projected on the graph. The active learning not only needs to reason about the current belief of the world, but belief posteriors conditioned on possible outcomes of edge evaluation.

(b) Explicitly reasoning about a set of paths is expensive as the size of the set can be exponential in the number of edges in the graph. An alternate method is to directly reason about a distribution over all possible paths between two vertices implicitly,

however, this can be intractable. Tractable approximations to such functions have been explored in the context of edge selection [9]. Adopting such techniques in the active learning setting would be interesting to pursue.

Q 2. *The active learning algorithms we use are restrictive and expensive.*
(a) Are there alternatives to solving DRD that are less restrictive?
(b) Are there more efficient approaches that do not require enumerating worlds?

(a) The restriction in DIRECT arises from the need to show *adaptive submodularity* for the surrogate objective. This is difficult to show in general as it depends not only on the objective but also on the set of possible realizations and the probability distribution over these realizations. In contrast, *interactive submodularity* [13], which only requires the objective be submodular for a fixed hypothesis, is easier to show and can lead to simpler surrogate functions.

(b) While DIRECT solves the exploration-exploitation problem in a principled fashion, it requires enumerating all plausible worlds and reasoning about them jointly. An alternative approach to efficient exploration-exploitation is via *posterior sampling for reinforcement learning (PSRL)* [21]. PSRL samples a *single* world from the posterior at the start of the episode, solves for an optimal policy and executes it. Hence PSRL efficiently exploits while exploration is automatically obtained by the variance of sampled worlds. Such policies also enjoy Bayesian regret bounds.

Q 3. *We have so far been concerned with finding a feasible path*
(a) Can we extend our framework to the optimal path identification problem?
(b) Furthermore, can we achieve asymptotic optimality via incremental densification?

(a) Introducing an additional criteria of minimizing path cost creates a tension between producing high quality paths and expending more evaluation effort. A desirable behaviour is to have an anytime algorithm that traverses the Pareto-frontier [6]. We can tweak our algorithm to display such behavior - we first solve the feasible path identification problem, prune all costlier paths (including this) from the library, prune worlds which belonged only to those paths, and then solve the feasible path problem again. However, while this will eventually converge to the optimal path, we can not necessarily control the speed of convergence.

(b) A naive approach to achieve asymptotic optimality is to add a batch of edges (densify) to the current graph, find the optimal path and repeat [7]. An key question is - where should we add samples? We would like to use the current belief about the world to actively add edges in promising areas that lead to discovery of better paths with little evaluation effort. Additionally, we would also like to interleave planning and densification such that the output of one constantly informs the other.

References

1. Robert Bohlin and Lydia E Kavraki. Path planning using lazy prm. In *ICRA*, 2000.
2. Brendan Burns and Oliver Brock. Sampling-based motion planning using predictive models. In *ICRA*, 2005.
3. John Canny. *The complexity of robot motion planning*. MIT press, 1988.
4. Yuxin Chen, Shervin Javdani, Amin Karbasi, Drew Bagnell, Siddhartha Srinivasa, and Andreas Krause. Submodular surrogates for value of information. In *AAAI*, 2015.
5. Sanjiban Choudhury, Shervin JAvdani, Siddhartha Srinivasa, and Sebastian Scherer. Near-optimal edge evaluation in explicit generalized binomial graphs. *Arxiv*, 2017.
6. Shushman Choudhury, Christopher M Dellin, and Siddhartha S Srinivasa. Pareto-optimal search over configuration space beliefs for anytime motion planning. In *IROS*, 2016.
7. Shushman Choudhury, Oren Salzman, Sanjiban Choudhury, and Siddhartha S Srinivasa. Densification strategies for anytime motion planning over large dense roadmaps. In *ICRA*, 2017.
8. Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *NIPS*, 2004.
9. Christopher M Dellin and Siddhartha S Srinivasa. A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors. In *ICAPS*, 2016.
10. Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Batch Informed Trees: Sampling-based optimal planning via heuristically guided search of random geometric graphs. In *ICRA*, 2015.
11. Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 2011.
12. Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal bayesian active learning with noisy observations. In *NIPS*, 2010.
13. Andrew Guillory and Jeff A. Bilmes. Interactive submodular set cover. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.
14. Jinwook Huh and Daniel D Lee. Learning high-dimensional mixture models for fast collision detection in rapidly-exploring random trees. In *ICRA*, 2016.
15. Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, Drew Bagnell, and Siddhartha Srinivasa. Near optimal bayesian active learning for decision making. In *AISTATS*, 2014.
16. Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.
17. S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006.
18. Steven M LaValle and James J Kuffner Jr. Randomized kinodynamic planning. *IJRR*, 2001.
19. Venkatraman Narayanan and Maxim Likhachev. Heuristic search on graphs with existence priors for expensive-to-evaluate edges. In *ICAPS*, 2017.
20. Christian L Nielsen and Lydia E Kavraki. A 2 level fuzzy prm for manipulation planning. In *IROS*, 2000.
21. Ian Osband, Dan Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *NIPS*, 2013.
22. Jia Pan, Sachin Chitta, and Dinesh Manocha. Faster sample-based motion planning using instance-based learning. In *WAFR*. Springer Verlag, 2012.
23. Fabio Ramos and Lionel Ott. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *IJRR*, 2016.
24. Abhijeet Tallavajhula, Sanjiban Choudhury , Sebastian Scherer, and Alonzo Kelly . List prediction applied to motion planning. In *ICRA*, 2016.