Planning with Affordances

Abstract

We introduce a novel approach to planning that combines the concept of affordances [?] with a standard planning algorithm, Value Iteration. Classical planning algorithms suffer from combinatoric state-space explosions [?] that cripple their effectiveness. Thus, in order to fully realize the potential of planning algorithms, we have sought to make these previously difficult problems more tractable; through the use of our affordance framework we seek to "guide" the agent as it plans, significantly reducing the size of exponential state-spaces. To accomplish this, we propose a planning algorithm that prunes the state action space using affordances. Evaluation is performed in the Minecraft domain on several path planning tasks - we demonstrate a significant increase in speed and reduction in state-space exploration across 5 different path planning tasks.

1 INTRODUCTION

As robots move out of the lab and into the real world, planning algorithms need to be able to scale to domains of increased noise, size, and complexity. In Norvig and Russell's AI textbook, they state that, "Planning is foremost an exercise in controlling combinatorial explosion." [?], an explosion which is currently uncontrolled across many planning domains. A classic formalization of this issue is the sequential decision making problem, where increases in problem size and complexity directly correspond to an explosion in the state-action space. Current approaches to solving sequential decision making problems cannot tackle these problems as the state-action space becomes large [?].

There is a strong need for a generalizable form of knowledge that, when coupled with a planner, is capable of solving problems in these massive domains. Humans provide an excellent existence proof for such planning, as we are capable of searching over an immense number of possible actions when presented with a goal - a recent concept out of 20th century psychological literature, an "affordance" has provided an explanation as to how this mechanism of human planning works.

Broadly, an affordance is "what [the environment] offers [an] animal, what [the environment] provides or furnishes, either for good or ill" [?], and may be employed in the context of planning to direct an agent toward relevant actions or parts of the environment (based on the agent's desires/goals). Additionally, roboticists have recently become interested in leveraging affordances for a variety of applications [?] [?]. In this paper we will employ affordances as a means of reducing large and intractable search problems into smaller, more reasonable problems. When instantiated in a physical robot, this will enable the robot to handle tasks that involve high order action spaces and expansive domains.

We propose a new formalism for representing affordances in an operationalized knowledge base. Current approaches to scaling MDPs up to large domains involve either pruning the state space of the world (by using subgoals), or limiting the action space [?]. In our approach, by introducing the affordance knowledge, we effectively limit the state space *and* the action space.

2 BACKGROUND

2.1 AFFORDANCES

An affordance is what an environment, or the objects in it, offer the agent. For example, if the agent were trying to drink coffee then a mug would afford carrying liquid. However if instead the agent wanted to secure some papers outside on a windy day, then the mug could afford weighing the papers down (i.e. affords using it as a paper weight). Often, objects may be defined in terms of what they afford, such as a bridge - suppose there is a log lying horizontally across a river; such a log provides passage over a river and could reasonably be referred to as a "bridge" simply because it affords crossing the river.

2.2 SUBGOALS

2.3 OPTIONS

2.4 MACRO-ACTIONS

2.5 OO-MDP

3 AFFORDANCE FORMALISM

An Affordance is defined as: $\mathcal{A} = <$ a, p, $\mathcal{G} >$, where:

$$a \in A$$

$$p \ : s \to \{0, 1\}$$

$$\mathcal{G} \ : s \to \{0, 1\}$$

Where a is a *primitive action* in the agent's set of available actions, p is a *precondition* which is a predicate on a given state s, and $\mathcal{G}$ is a desired *postcondition*, which is also a predicate on a particular state s.

4 EXPERIMENTS

4.1 BASELINES

5 RESULTS