Toward Affordance-Aware Planning

David Abel & Gabriel Barth-Maron
James MacGlashan, Stefanie Tellex

Dept. of Computer Science, Brown University

RSS: First Workshop on Affordances
July 13, 2014
Robotics Motivation

Tool Use, Subgoal Planning

Problem Statement

Enable Autonomous Agents to learn to plan effectively in massive stochastic state-spaces.
Minecraft & Robotics


http://youtu.be/fSLh92zCgIg?t=1m12s

[3] Ross A. Knepper, Todd Layton, John Romanishin, and Daniela Rus
Minecraft

https://vimeo.com/99896931

≈Turing Complete LEGO
Minecraft: The Problem

\[ \approx \text{State Space Size: } 10^{181} \]

**ACTIONS**

- Move
- Place
- Destroy
- Use
- Jump
- Rotate
- Look
- Craft
- ...
Affordances In Planning

- walk
- build bridge
- push
- pull
Affordances as knowledge given to an MDP

Idea: Affordances focus the agent on relevant action possibilities by pruning irrelevant actions on a state by state basis.
We define an affordance as:

$$\Delta = \langle p, g \rangle \rightarrow \mathcal{A}'$$

Where:

- $\Delta$ = symbol for an affordance
- $p$ = predicate on states
- $g$ = lifted goal description
- $\mathcal{A}'$ = subset of MDP actions
Affordances Example

\[
\langle \text{nearTrench, atGoal} \rangle \rightarrow \{\text{place, jump}\}
\]

\[
\langle \text{near Plane, atGoal} \rangle \rightarrow \{\text{move, rotate}\}
\]
Affordances Example: State-Action Space Pruning

\[ \langle \text{nearTrench, atGoal} \rangle \mapsto \{\text{place, jump}\} \]
\[ \langle \text{nearPlane, atGoal} \rangle \mapsto \{\text{move, rotate}\} \]
Sample Results: Expert Affordances

Avg. # Bellman Updates to solve OO-MDP on Test State Space

- Value Iteration
- RTDP
- Subgoal Planning

Bellman Updates

- Regular
- Affordance-Aware
Demo

https://vimeo.com/88689171
Learning Framework: Goal

1. Compute the probability that an action set contains the optimal action for each reachable state in the MDP

2. Learn Dirichlet priors on action sets for each affordance to inform this distribution
Learning Framework

1. Compute the probability that an action set contains the optimal action for each reachable state in the MDP

\[
\Pr(\mathcal{A}^* \mid s, \Delta_1, \ldots, \Delta_K) = \Pr(\mathcal{A}'_1 \cup \ldots \cup \mathcal{A}'_K \mid s, \Delta_1, \ldots, \Delta_K) \\
\approx \sum_{i=1}^{K} \Pr(\mathcal{A}'_i \mid s, \Delta_i)
\]

*assume action sets are disjoint

Key:

\[
\begin{align*}
\mathcal{A}^* & = \text{candidate set of actions} \\
\Delta_i & = \text{the } i\text{-th affordance} \\
s & = \text{MDP state} \\
\mathcal{A}'_i & = \text{the } i\text{-th affordance’s action set}
\end{align*}
\]
Learning Framework: Goal

2. **Learn Dirichlet priors on action sets for each affordance that inform this distribution**

\[
\Pr(\mathcal{A}^* \mid s, \Delta_1, \ldots, \Delta_K) \approx \sum_{i=1}^{K} \Pr(\mathcal{A}'_i \mid s, \Delta_i)
\]

\[
\Pr(\mathcal{A}'_i \mid s, \Delta_i) = \Pr(\mathcal{A}'_i \mid n_i, \lambda_i) = \Pr(\lambda_i \mid \alpha_i) \cdot \Pr(n_i \mid \beta_i)
\]

**Where:**

\[
\Pr(\lambda_i \mid \alpha_i) = \text{DirMult}(\alpha_i) \quad \text{and} \quad \Pr(n_i \mid \beta_i) = \text{Dir}(\beta_i)
\]

**Key:**

- \(\mathcal{A}^*\) = candidate set of actions
- \(s\) = MDP state
- \(\lambda_i\) = multinomial over actions
- \(\alpha_i\) = Dirichlet parameters
- \(\Delta_i\) = the \(i\)-th affordance
- \(\mathcal{A}'_i\) = the \(i\)-th affordance’s action set
- \(n_i\) = multinomial over action set size
- \(\beta_i\) = Dirichlet Multinomial parameters
Full Learning Process

Input: \( P \) (set of predicates), \( G \) (set of goal descriptions)

RANDOMLY GENERATE SMALL WORLDS:

SOLVE EACH MDP

COUNT:

\[ \Delta_i \cdot \text{parameterUpdate}(\pi_1, \ldots, \pi_N) \]
Learning Example: Before Learning

\( p = \text{nearTrench} \)
\( g = \text{AtGoal} \)

Multinomial Over Action Set Size

Multinomial Over Actions

\( \text{move} \quad \text{rotateR} \quad \text{rotateL} \quad \text{destroy} \quad \text{jump} \quad \text{place} \quad \text{lookD} \quad \text{lookU} \quad \text{use} \)
Learning Example: After Learning

\[ p = \text{nearTrench} \]
\[ g = \text{AtGoal} \]

\[ \text{Multinomial Over Action Set Size} \]

\[ \text{Multinomial Over Actions} \]

- move
- rotateR
- rotateL
- destroy
- jump
- place
- lookD
- lookU
- use
Learning Example

Multinomial Over Action Set Size

Sample $n$

Multinomial Over Actions

Take $n$ unique samples $A$
Learning Results

Average # Bellman Updates Per Converged Policy

- No Affordances
- With Learned Affordances
- With Expert Affordances

Tiny World
Small World
Medium World
Large World
Related Work

• Temporarily Extended Actions [5, 6, 7]

• Temporal Logic [8, 9]

• Action Pruning [11, 12]

• Heuristics [13, 14]
Future Work

• Extending predicates to logical expressions
• Incorporate more of the Minecraft Domain
• Learning High-level Representations
• Deploy on robots, other domains (cooking, javascript, Atari)
Contributions

• Defined affordance as formal knowledge added to an MDP.

• Realized speedups for planning in large stochastic state spaces.

• Demonstrated framework for learning affordances from interaction in the world.
References


[2] https://www.youtube.com/watch?v=B9sYogRVF8Q


[4] http://qph.is.quoracdn.net/main-qimg-f0a341a110341f5a58a93b75b491448d


Probability that an action set contains the optimal action for each reachable state in the MDP:

\[
\Pr(\mathcal{A}^* \mid s, \Delta_1, \ldots, \Delta_K)
\]

approximate assuming action sets are disjoint:

\[
\Pr(\mathcal{A}^* \mid s, \Delta_1, \ldots, \Delta_K) \approx \sum_{i=1}^{K} \Pr(\mathcal{A}'_i \mid s, \Delta_i)
\]

For each affordance:

\[
\Pr(\mathcal{A}'_i \mid s, \Delta_i) = \Pr(\mathcal{A}'_i \mid n_i, \lambda_i) = \Pr(\lambda_i \mid \alpha_i) \cdot \Pr(n_i \mid \beta_i)
\]

Key:

\[
\begin{align*}
\mathcal{A}'_i &= \text{Affordance}_i \text{ action set} \\
\mathcal{A}^* &= \text{Candidate action set} \\
\lambda &= \text{Actions sampled from affordance-specific multinomial} \\
n &= \text{Action set size sampled from affordance-specific multinomial} \\
\Delta_i &= \text{Affordance}_i \\
\alpha &= \text{Dirichlet Multinomial parameters for prior on actions} \\
\beta &= \text{Dirichlet parameters for prior on action set size} \\
s &= \text{MDP state}
\end{align*}
\]
Key:

\( A_i' \) = Affordance\(_i\) action set
\( A^* \) = Candidate action set
\( \lambda \) = Actions sampled from affordance-specific multinomial
\( n \) = Action set size sampled from affordance-specific multinomial
\( \Delta_i \) = Affordance\(_i\)
\( \alpha \) = Dirichlet Multinomial parameters for prior on actions
\( \beta \) = Dirichlet parameters for prior on action set size
\( s \) = MDP state