

Markovian State and Action Abstractions for MDPs via Hierarchical MCTS

Aijun Bai

March 31, 2016

UC Berkeley

State Abstraction

State Abstraction

- Ground MDP: $\langle S, A, T, R \rangle$
- Group a set of states as a unit
 - Abstract states:
 $X = \{x_1, x_2, \dots\}$
 - A partition on S
 - Abstraction function:
 $\phi : S \rightarrow X$
- Advantages:
 - Reduced abstract state space size
 - Reduced stochastic branching factors

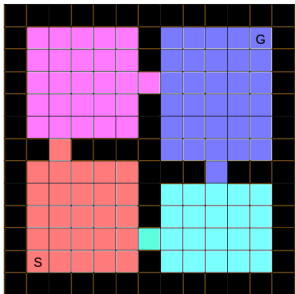


Figure 1: Rooms domain

Non-Markovianess

- The transition system in abstract state space is non-Markovian
 - $\Pr(x'|x, a) = \sum_{s' \in x'} \sum_{s \in x} T(s'|s, a) \Pr(s|x)$
 - Occupancy probability:
 $\Pr(s|x)$
 - Bayesian update:
 $\Pr(s'|hax) = \eta \mathbf{1}[\phi(s') = x] \sum_{s \in S} T(s'|s, a) \Pr(s|h)$
 - * Where, $h = x_0 a_0 \cdots x_n$
 - * Dependent on the history, or in other words, the policy being executed/computed!

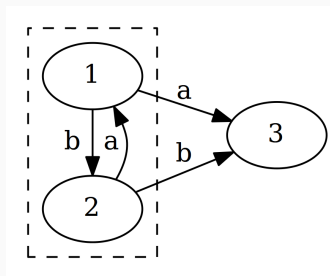


Figure 2: A 3-state MDP

State Abstraction in the Literature

- Safe abstraction
 - Ignore only irrelevant state variables (Dietterich, 1999; Andre & Russell, 2002)
 - Exploit particular symmetric structure (e.g. bisimulation or homomorphism) in the transition function (Dearden & Boutilier, 1997; Givan et al., 2003; Jiang et al., 2014; Anand et al., 2015)
 - Not always possible

State Abstraction in the Literature (cont'd)

- Weighting function/aggregation probability:
 $w(s, x) \approx \Pr(s|x)$ (Bertsekas et al., 1995; Singh et al., 1995; Li et al., 2006; Hostetler et al., 2014)

$$- T_\phi(x'|x, a) = \sum_{s' \in x'} \sum_{s \in x} T(s'|s, a) w(s, x)$$

$$- R_\phi(x, a) = \sum_{s \in x} R(s, a) w(s, x)$$

- Abstract MDP: $\langle X, A, T_\phi, R_\phi \rangle$
- Abstract policy: $\pi_\phi : X \rightarrow A$
- Problem: $\Pr(s|x)$ is non-stationary, which can not be well approximated by a constant weighting function

State Abstraction from a POMDP Perspective

- State abstraction introduces partial observability
 - Abstract states as observations
- Doing state abstraction ϕ over an MDP $M = \langle S, A, R, T \rangle$ creates a POMDP: $M|_{\phi} = \langle S, A, X, T, R, \Omega \rangle$
 - Observation function: $\Omega(x|s) = \mathbf{1}[x = \phi(s)]$
 - Belief state $b(s)$ gives the occupancy probability

Weighting Function from a POMDP Perspective

- Approximate belief state using a constant distribution $w(s, x)$ for each observation x
- Find a memory-less policy $\pi_\phi : X \rightarrow A$
 - Let π be the optimal policy for M
 - Let $\pi_{M|\phi}$ be the optimal policy for $M|\phi$
 - Performance: $\pi_\phi \prec \pi_{M|\phi} \prec \pi$
- The weighting function approach doesn't seem to be well motivated!

Overcome Non-Markovianess via a POMDP Formulation

- Exactly solving $M|_{\phi}$ via dynamic programming is intractable
 - Continuous belief space
- The search tree in $M|_{\phi}$ has lower branching factor than in M
 - Let $\mathcal{T}(s_0)$ be the search tree in M starting from s_0
 - Let $\mathcal{T}(b_0)$ be the search tree in $M|_{\phi}$ starting from b_0 , where $b_0(s) = \mathbf{1}[s = s_0]$
 - Branching factor: $|A||X| \ll |A||S|$
- Solving $M|_{\phi}$ via point-based method is possible
 - Find the near-optimal action for b_0 by building an expectimax tree

Solving $M|_{\phi}$ via Monte Carlo Tree Search

- $MCTS|_{\phi}$: run POMCP in $M|_{\phi}$
 - Build a search tree in the history space via sampling
 - Use particles to approximate the belief at the root node
 - * Not necessary, since the ground state at the root node can be observed
- Advantages of $MCTS|_{\phi}$:
 - Only a simulator for the ground MDP is needed
 - Can be applied to continuous state-space MDPs

Action Abstraction

Action Abstraction in $M|_\phi$

- What we have now:
 - $MCTS|_\phi$: a MCTS algorithm with Markovian state abstraction for MDPs via a POMDP formulation
- Action abstraction decomposes the overall problem into a hierarchy of sub-problems
 - Abstract actions/options/subtasks/HAMs
- The transitions between abstract states forms a natural hierarchical structure in $M|_\phi$
 - Abstract actions as transitions between abstract states

Value Function Decomposition

- Let $\alpha \in \mathcal{A}$ be an abstract action: $\alpha = \langle x \in X, y \in X, A, \pi \rangle$
- Let π be a hierarchical policy: $\pi = \{\pi_0, \pi_1, \pi_2, \dots\}$
 - Where, $\pi_0 : \mathcal{H} \rightarrow \mathcal{A}$ is the root task
- Hierarchical decomposition:
 - $V(\pi, h) = Q(\pi, h, \pi_0(h))$
 - $Q(\pi, h, \alpha) =$
 $V(\alpha, h) + \sum_{h' \in \mathcal{H}} \gamma^{|h'| - |h|} \Pr(h'|h, \alpha) V(\pi, h')$
 - $V(\alpha, h) = Q(\alpha, h, \pi_\alpha(h))$
 - $Q(\alpha, h, a) = R(h, a) + \gamma \sum_{x \in X} \Pr(x|h, a) V(\alpha, hax)$

- Executing an abstract action takes a random number of steps
 - $N = |h'| - |h|$
 - $h' \sim \text{Pr}(h'|h, \alpha)$ — terminating distribution of α
 - Terminating distributions are unknown in advance
 - * They are dependent on local policies
 - * Local policies, as well as the high-level policy, are being computed!

Overcome Semi-Markovianess via Monte-Carlo Learning

- High-level:
 - Learn π_0 by running MCTS over abstract actions
- Low-level:
 - Learn π_α by running MCTS over primitive actions
 - * π_α converges given infinite samples
 - * $\Pr(h'|h, \alpha)$ also converges as π_α converges
- Action values are backuped according to the hierarchical decomposition
- The hierarchical policy π converges to a recursively optimal hierarchical policy

The Overall Algorithm

- $\text{MCTS}|_{\phi, \mathcal{A}}$: a MCTS algorithm with Markovian state and action abstractions for MDPs via a POMDP formulation
 - Overcome non-Markovianess resulting from doing state abstraction via a POMDP formulation
 - Overcome semi-Markovianess resulting from doing action abstraction via a Monte Carlo learning process

Theoretical Results

Aggregation Error

- The aggregation error of state abstraction ϕ is e , if for all $x \in X$ and $s \in x$, $\exists \hat{a} \in A$, such that $V^*(s) - Q^*(s, \hat{a}) \leq e$, where V^* and Q^* are optimal value and action-value functions for the ground MDP M
 - $e = 0$ implies that all ground states within the same abstract state share the same optimal action
- The bounded aggregation error requires that the action value $Q^*(s, \hat{a})$ of \hat{a} is close to the optimal value $V^*(s)$ for all ground states $s \in x$ within abstract state x
 - Measures the quality of the state abstraction

Optimality Results with State Abstraction

Theorem

(In short) The performance of the optimal policy $\pi_{\mathcal{M}|\phi}$ of $\mathcal{M}|\phi$ is bounded by a constant multiple of the state aggregation error, comparing with the optimal policy π of \mathcal{M}

Convergence Results with Action Abstraction

Theorem

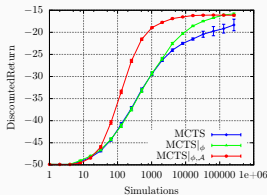
$MCTS|_{\phi, \mathcal{A}}$ converges to a recursively optimal hierarchical policy for $M|_{\phi}$ over the hierarchy defined by abstract actions with probability 1

Empirical Evaluation

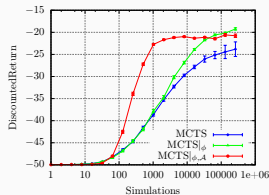
The Rooms Domain

- The ROOMS[m, n, k] problem:
 - A robot navigates in a $m \times n$ grid map containing k rooms
 - Primitive actions: E, S, W, N
 - Probability 0.2 of moving into perpendicular positions
 - Abstract states: rooms
 - Abstract actions: transitions between rooms

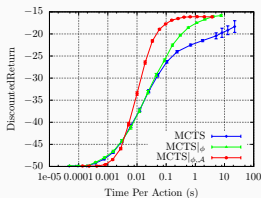
Experimental Results



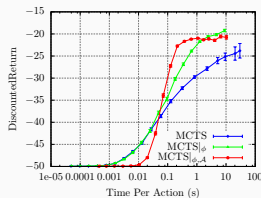
(a) ROOMS[17, 17, 4]



(b) ROOMS[25, 13, 8]



(c) ROOMS[17, 17, 4]



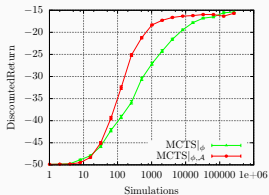
(d) ROOMS[25, 13, 8]

Figure 3: Empirical results on the rooms rooms domain.

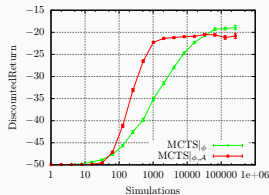
The Continuous Rooms Domain

- The C-ROOMS[m, n, k, w] problem:
 - Each grid has a size of $w \times w$ (m^2)
 - The position of the robot: continuous (x, y)
 - Primitive actions: E, S, W, N
 - * Movement is augmented with a Gaussian error
 - * Move in stochastic directions by a distance of w in expectation
 - MCTS in such continuous domain reduces to depth-1 search

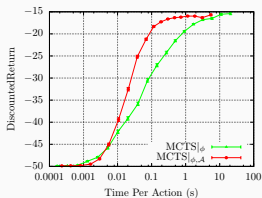
Experimental Results



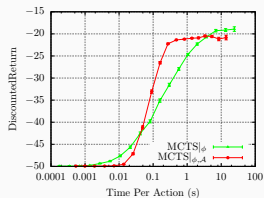
(a)
C-ROOMS[17, 17, 4, 1]



(b)
C-ROOMS[25, 13, 8, 1]



(c)
C-ROOMS[17, 17, 4, 1]



(d)
C-ROOMS[25, 13, 8, 1]

Conclusion

Conclusion

- A hierarchical MCTS algorithm with Markovian state and action abstractions for MDPs
 - Overcome non-Markovianess introduced by state abstraction via a POMDP formulation
 - Overcome semi-Markovianess introduced by action abstraction via a Monte Carlo learning process
 - Find a recursively hierarchical optimal policy bounded by a multiple constant of an aggregation error

References

- Anand, A., Grover, A., Mausam, M., & Singla, P. (2015). ASAP-UCT: abstraction of state-action pairs in UCT. In *Proceedings of the 24th International Conference on Artificial Intelligence*, (pp. 1509–1515). AAAI Press.
- Andre, D., & Russell, S. J. (2002). State abstraction for programmable reinforcement learning agents. In *AAAI/IAAI*, (pp. 119–125).
- Bertsekas, D. P., Bertsekas, D. P., Bertsekas, D. P., & Bertsekas, D. P. (1995). *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA.
- Dearden, R., & Boutilier, C. (1997). Abstraction and approximate decision-theoretic planning. *Artificial Intelligence*, 89(1), 219–283.
- Dietterich, T. G. (1999). State abstraction in MAXQ hierarchical reinforcement learning. *arXiv preprint cs/9905015*.
- Givan, R., Dean, T., & Greig, M. (2003). Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1), 163–223.
- Hostetler, J., Fern, A., & Dietterich, T. (2014). State aggregation in Monte Carlo tree search. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.
- Jiang, N., Singh, S., & Lewis, R. (2014). Improving UCT planning via approximate homomorphisms. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, (pp. 1289–1296). International Foundation for Autonomous Agents and Multiagent Systems.
- Li, L., Walsh, T. J., & Littman, M. L. (2006). Towards a unified theory of state abstraction for MDPs. In *ISAIM*.
- Singh, S. P., Jaakkola, T., & Jordan, M. I. (1995). Reinforcement learning with soft state aggregation. *Advances in neural information processing systems*, (pp. 361–368).

The Pseudo Code

```

Agent ( $s_0$  : initial ground state)
 $h \leftarrow \emptyset$ 
 $\mathcal{P}(h) \leftarrow \{s_0\}$ 
repeat
   $\mathcal{T} \leftarrow$  an empty tree
   $a \leftarrow \text{OnlinePlanning}(h)$ 
  Execute  $a$  and observe abstract state  $x$ 
   $h \leftarrow \text{hax}$ 
   $\mathcal{P}(h) \leftarrow \text{ParticleFilter}(\mathcal{P}(h), a, x)$ 
until terminating conditions

Rollout ( $\alpha$  : task,  $s$  : state,  $h$  : history,  $d$  : depth)
if  $d \geq H$  or  $\alpha$  terminates at  $h$  then
   $\text{return } (0, 0, h, s)$ 
else
   $a \leftarrow \text{GetPrimitive}(\pi_{\text{rollout}}, \alpha, h)$ 
   $(s', x, r') \leftarrow \text{Simulate}(s, a)$ 
   $(r'', n, h'', s'') \leftarrow \text{Rollout}(\alpha, s', \text{hax}, d + 1)$ 
   $r \leftarrow r' + \gamma r''$ 
   $\text{return } (r, n + 1, h'', s'')$ 

GetGreedyPrimitive ( $\alpha$  : task,  $h$  : history)
if  $\alpha$  is primitive then
   $\text{return } \alpha$ 
else
   $a^* \leftarrow \arg\max_a Q(\alpha, h, a)$ 
   $\text{return GetGreedyPrimitive}(a^*, h)$ 

GetPrimitive ( $\pi$  : policy,  $\alpha$  : task,  $h$  : history)
if  $\alpha$  is primitive then
   $\text{return } \alpha$ 
else
   $\text{return GetPrimitive}(\pi, \pi_\alpha(h), h)$ 

OnlinePlanning ( $h$  : history)
repeat
   $s \sim \mathcal{P}(h)$ 
  Search (root task,  $s, h, 0$ )
until resource budgets reached
return GetGreedyPrimitive (root task,  $h$ )

Search ( $\alpha$  : task,  $s$  : state,  $h$  : history,  $d$  : depth)
if  $\alpha$  is primitive then
   $(s', x, r) \sim \text{Simulate}(s, \alpha)$ 
   $\text{return } (r, 1, \text{hax}, s')$ 
else
  if  $d \geq H$  or  $\alpha$  terminates at  $h$  then
     $\text{return } (0, 0, h, s)$ 
  else
    if node  $\langle \alpha, h \rangle$  is not in tree  $\mathcal{T}$  then
      Insert node  $\langle \alpha, h \rangle$  to  $\mathcal{T}$ 
       $\text{return Rollout}(\alpha, s, h, d)$ 
    else
       $a^* \leftarrow \arg\max_a \left\{ \hat{Q}(\alpha, h, a) + c \sqrt{\frac{\log N(\alpha, h)}{N(\alpha, h, a)}} \right\}$ 
       $(r', n', h', s') \leftarrow \text{Search}(a^*, s, h, d)$ 
       $(r'', n'', h'', s'') \leftarrow \text{Search}(\alpha, s', h', d + n')$ 
       $N(\alpha, h) \leftarrow N(\alpha, h) + 1$ 
       $N(\alpha, h, a^*) \leftarrow N(\alpha, h, a^*) + 1$ 
       $r \leftarrow r' + \gamma n' r''$ 
       $\tilde{r} \leftarrow r' + \gamma n' r'' + \gamma n' + n'' \hat{R}_\alpha(h'')$ 
       $Q(\alpha, h, a^*) \leftarrow Q(\alpha, h, a^*) + \frac{r - Q(\alpha, h, a^*)}{N(\alpha, h, a^*)}$ 
       $\hat{Q}(\alpha, h, a^*) \leftarrow \hat{Q}(\alpha, h, a^*) + \frac{\tilde{r} - \hat{Q}(\alpha, h, a^*)}{N(\alpha, h, a^*)}$ 
       $\text{return } (r, n' + n'', h'', s'')$ 

```

Figure 5: $\text{MCTS}|_{\phi, \mathcal{A}}$ — Monte-Carlo tree search with state and action abstractions for MDPs

Optimality Results

Theorem

For state abstraction $\langle X, \phi \rangle$ with aggregation error e , let s_0 be the current state in the ground MDP M and h_0 with $\mathcal{P}(h_0) = \{s_0\}$ be the corresponding history node in POMDP $M|_\phi$. Let $Q^(s, \cdot)$ and $Q^*(h, \cdot)$ be the optimal action values for M and $M|_\phi$ respectively. Let $\alpha^* = \operatorname{argmax}_{\alpha \in A} Q^*(h_0, \alpha)$ be the optimal action found in $M|_\phi$ at history h_0 , and define action-value error as $E(\alpha^*) = |\max_{\alpha \in A} Q^*(s_0, \alpha) - Q^*(s_0, \alpha^*)|$. Suppose the maximal planning horizon is H , then $E(\alpha^*)$ is bounded by $E(\alpha^*) \leq 2He$ if $\gamma = 1$, else $E(\alpha^*) \leq 2\gamma \frac{1-\gamma^H}{1-\gamma} e$.*