On the Hardness of Finding Symmetries in Markov Decision Processes

Shravan M Narayanamurthy\(^1\) Balaraman Ravindran\(^2\)

\(^1\)Yahoo! Labs
Bangalore

\(^2\)Dept. of Computer Science and Eng.
Indian Institute of Technology Madras

July 6\(^{th}\) / ICML’08
Markov Decision Processes (MDPs)
- Used to model sequential decision problems
- Current solution techniques do not scale well with the size of the MDP
- Real world problems when modeled as MDPs exhibit high degree of redundancy
- Reduction in size possible if we exploit redundancy

Finding Symmetries in MDPs
- We use the symmetry as a notion of redundancy as introduced in (Ravindran, 2004)
- Believed to be hard however exact hardness is unknown
- Intuitively, because of the additional structure of MDPs it seems harder

We show that finding symmetries in MDPs is no harder than the problem of Graph Isomorphism (GI)

We also show the use of existing GI solvers for finding symmetries in MDPs
Stochastic Sequential Decision Making

- Markov Decision Process, $\mathcal{M} : < S, A, \Psi, P, R >$
  - Set of States : $S$
  - Set of Actions : $A$
  - Set of Permissible Actions : $\Psi \subseteq S \times A$
  - Transition Probabilities : $P : \Psi \times S \rightarrow [0, 1]$
  - Expected Reward : $R : \Psi \rightarrow \mathbb{R}$

- Policy, $\pi : S \rightarrow A$

- Value of a state $s$ under policy $\pi : E^\pi$ (discounted sum of future rewards got by following $\pi$ from $s$)

- Bellman Equation

$$V_\pi(s) = R(s, a) + \gamma \sum_{s' \in S} P(s, a, s') V_\pi(s')$$

where, $0 \leq \gamma < 1$
Solution of an MDP

- Is a policy $\pi^*$ such that, for any policy $\pi$, $V_{\pi^*}(s) \geq V_{\pi}(s)$, $\forall s \in S$
- Bellman Optimality Equation

$$V_{\pi^*}(s) = \max_{a \in A} \{ R(s, a) + \gamma \sum_{s' \in S} P(s, a, s') V_{\pi^*}(s') \}$$

- Iterative algorithm using the Bellman Optimality equation
Outline

1. Overview
2. Introduction
   - Markov Decision Processes
   - Formal Problem Definition
3. Symmetries in MDPs
   - Finding Symmetries
   - Exploiting Symmetries
4. Experiments and Results
5. Conclusions
Reduced Model - Formal definition

**Definition**

An MDP homomorphism from $\mathcal{M}$ to $\mathcal{M}'$ is a surjection $h : \Psi \rightarrow \Psi'$ defined by $h(s, a) = (f(s), g_s(a))$, where $f : S \rightarrow S'$ and $g_s : A_s \rightarrow A'_{f(s)}$ are surjections satisfying,

$$P'(f(s), g_s(a), f(s')) = \sum_{s'' \in f^{-1}(f(s'))} P(s, a, s'')$$

$$R'(f(s), g_s(a)) = R(s, a)$$

**Definition**

An MDP $\mathcal{M}'$ is said to be a *reduced model* of an MDP $\mathcal{M}$, iff there exists an MDP homomorphism $h : \mathcal{M} \rightarrow \mathcal{M}'$. 

Shravan M Narayanamurthy, Balaraman Ravindran

On the Hardness of Finding Symmetries in Markov Decision Processes
Reduced Model - Significance

Reduced Model:
- Preserves dynamics by definition
- Preserves optimal value functions and policies
- Functionally equivalent to the original model but significantly smaller
Symmetry informally

- A symmetric system is one that is invariant under certain transformation onto itself.
- This gridworld is invariant under reflection along diagonal
A bijective MDP homomorphism from $M$ to $M$ is called an MDP automorphism which represents a symmetry. We have,

\[ P(f(s), g_s(a), f(s')) = P(s, a, s') \]
\[ R(f(s), g_s(a)) = R(s, a) \]

The set of all automorphisms of an MDP, $M$, form a group under composition called the automorphism group of $M$, represented as $\text{Aut}M$. The orbits of the natural action of any subgroup $G$ on $M(\Psi)$ defines a partition $B_G$ of $\Psi$ using which a quotient MDP $M/B_G$, called the $G$-Reduced Image can be defined.
Problem

- Given an MDP $M$
  1. Find $\text{Aut}_M$
  2. Find $\text{Aut}_M$-Reduced Image
Outline

1 Overview

2 Introduction
   - Markov Decision Processes
   - Formal Problem Definition

3 Symmetries in MDPs
   - Finding Symmetries
   - Exploiting Symmetries

4 Experiments and Results

5 Conclusions
Problem Simplification

- Given an MDP $\mathcal{M}$, find $\text{Aut}\mathcal{M}$
- A group is completely specified by its generators
- $\text{AMGEN} (\mathcal{M})$: Find generators of $\text{Aut}\mathcal{M}$
Isomorphism Completeness

Definition

A is *Isomorphism Complete* iff A is polynomially equivalent to finding Graph Isomorphisms

Definition

A is *polynomially equivalent* to B iff A is polynomially reducible ($\propto$) to B and $B \propto A$, denoted $A \equiv_{\propto} B$
List of relevant Isomorphism Complete Problems

- ISO($G_1, G_2$): Isomorphism recognition for $G_1$ and $G_2$, where $G_1$ and $G_2$ are simple
- IMAP($G_1, G_2$): Isomorphism Map from $G_1$ to $G_2$ (if it exists), where $G_1$ and $G_2$ are simple
- AGEN($G$): Generators of the automorphism group, Aut$G$, where $G$ is simple
- DGEN($G$): Generators of the automorphism group, Aut$G$, where $G$ is a digraph
- So, $DGEN(G) \equiv_\infty AGEN(G) \equiv_\infty IMAP(G_1, G_2) \equiv_\infty ISO(G_1, G_2)$
Outline

1. Pose $AMGEN(\mathcal{M})$ as a problem on weighted pseudographs
2. Prove that $AMGEN(\mathcal{M}) \equiv_{\infty} DGEN(G)$
   - $DGEN(G) \propto AMGEN(\mathcal{M})$ (trivial)
   - $AMGEN(\mathcal{M}) \propto DGEN(G)$
3. Hence, $AMGEN(\mathcal{M})$ is Isomorphism Complete
Set Bijection

1. A generator of $\text{Aut}_\mathcal{M}$ has 2 components:
   - A function $f$ that permutes the states
   - A set of functions $\{g_u\}$ that permute the actions called the State-Dependent Action Recoding (SDAR) functions.

2. Solution to $DGEN(G)$ accounts only for $f$

3. Factorially many SDAR functions in the worst case, rendering explicit representations useless

4. To obtain the SDAR functions, we define the notion of a set bijection

5. Represents a set of bijections compactly

6. Polynomialsly computable operations of intersection, composition and inverse
Set Bijections example

For example, the following set of bijections from $A = \{1, 2, 3, 4\}$ to $B = \{N, E, W, S\}$

1 $\rightarrow$ N, 2 $\rightarrow$ E, 3 $\rightarrow$ W, 4 $\rightarrow$ S
1 $\rightarrow$ N, 2 $\rightarrow$ E, 3 $\rightarrow$ S, 4 $\rightarrow$ W
1 $\rightarrow$ E, 2 $\rightarrow$ N, 3 $\rightarrow$ W, 4 $\rightarrow$ S
1 $\rightarrow$ E, 2 $\rightarrow$ N, 3 $\rightarrow$ S, 4 $\rightarrow$ W

can be represented compactly using a set bijection, $X_1$, from $U^1_A = \\{\{1, 2\}, \{3, 4\}\}$ to $U^1_B = \\{\{N, E\}, \{W, S\}\}$ as follows:

$X_1(\{1, 2\}) = \{N, E\}$
$X_1(\{3, 4\}) = \{W, S\}$
Finding Symmetries

Shravan M Narayanamurthy, Balaraman Ravindran

On the Hardness of Finding Symmetries in Markov Decision Processes
An example

Vector Weighted Graph

Vector label: N,E,W,S
Discarded as it is same for each edge
On the Hardness of Finding Symmetries in Markov Decision Processes
An example

Weighted DiGraph (WG)

An automorphism in AutWG:

F: 1 2 3 4 5 6 7 8 9
1 4 7 2 5 8 3 6 9
(2 4)(3 7)(6 8)
Finding Symmetries

An example

Finding State Dependent Action
Recoding Functions

G_{uv} = Q_{f(u)}^{-1} \cdot Q_{f(v)}

G_u = \text{intersection}(G_{uv}) \text{ over all } v

G11: \{N,E\} - \{N,E\}, \{E\} - \{N\}
G14: \{E\} - \{N\}

G: \{N\} - \{E\}, \{E\} - \{N\}, \{W,S\} - \{W,S\}

F: 1 2 3 4 5 6 7 8 9
1 4 7 2 5 8 3 6 9

(2 4)(3 7)(6 8)
Construction provides the Generators of $\text{Aut}_M$

Outline of the proof

1. Prove that $\text{Aut}_M$ can be partitioned into $\{< f, \{G_u\} >\}$.
2. Define a group homomorphism $\phi : \text{Aut}_M \rightarrow \text{Aut}_{WG}$.
3. Prove that $\text{Aut}_M$ as partitioned above represents the set of all cosets of the kernel, $\ker(\phi)$.
4. Since, the kernel is a normal subgroup, we know that, $\text{Aut}_M / \ker(\phi) \cong \text{im}(\phi)$.
5. Using the isomorphism, prove that the set $\{< f, \{G_u\} >\}$ found using the above procedure is the set of generators of $\text{Aut}_M$. 
Significance

1. Theoretically significant
2. Allows the use of off-the-shelf Graph Isomorphism solvers to find symmetries on MDPs.
Nauty - No Automorphisms, Yes?

- Solves DGEN
  - Worst case complexity is exponential
  - On avg on a graph of $n$ vertices takes $n^2$ time
- Uses backtracking and a refinement procedure to find the canonical labelings
- Allows the use of a variety of vertex invariants to solve harder problems
Nauty Integration

Procedure

1. Construct the weighted pseudograph from the given MDP
2. Construct the weighted digraph using the above procedure
3. Construct a simple digraph from the weighted digraph using standard procedure
4. Get the generators of the digraph using Nauty
5. Use set bijections to find state-dependent action recoding functions for each generator
6. Generate the partition of $|\Psi|$ induced by the group generated by the above functions
7. Use the partition to construct a reduced model and follow explicit model minimization
Outline

1. Overview

2. Introduction
   - Markov Decision Processes
   - Formal Problem Definition

3. Symmetries in MDPs
   - Finding Symmetries
   - Exploiting Symmetries

4. Experiments and Results

5. Conclusions
Given an MDP $\mathcal{M}$ and a Symmetry Group $\mathcal{G}$, finds the reduced model $\mathcal{M}'$ induced by $\mathcal{G}$

- Straightforward way by explicit enumeration takes $|\Psi| \times |\mathcal{G}|$
- Breadth First Search with pruning
- Terminates when at least one representative from each equivalence class of $\mathcal{G}$ has been examined
- With certain assumptions time complexity is $O(|\Psi'| \times |\mathcal{G}|)$
Experimental Setup - Probabilistic GridWorld

- **States**: An $N \times N$ GridWorld
- **Actions**: Four probabilistic actions of going UP, DOWN, RIGHT and LEFT having a 90% success probability
- **Initial state**: $(0,0)$
- **Goal states**: $\{(0, N-1), (N-1, 0)\}$. 
**Experimental Setup - GridWorld Soccer**

- Slightly modified version of that described in (Bowling, 2003) with an $M \times N$ grid with two agents (Attacker-A and Defender-B)

- **States:** The non-identical positions of the attacker and the defender leading to $(MN)^2 - (MN)$ states

- **Actions:** The four compass directions: N, E, W, S and the hold action H

- **Goal States:** W action from the squares in front of the goal
Able to find the partition corresponding to the symmetry group

For a grid of size $N \times N$, states $(x,y)$, $(y,x)$, $(N-1-x,N-1-y)$ and $(N-1-y,N-1-x)$ are equivalent
Intuition gets it wrong; domain is not symmetric!

The algorithm also finds another interesting symmetry due to the existence of the hold action.
In this work, we have provided a constructive proof for the *Isomorphism Completeness* of the problem of finding symmetries. We have also proposed the use of this constructive proof along with an efficient minimization algorithm to solve an MDP using symmetries and demonstrated it empirically.

We are looking at adapting approximation algorithms for finding graph isomorphisms to finding approximate symmetries in MDPs.
Thank You!
Given $\mathcal{M} = \langle S, A, \Psi, P, R \rangle$ and $\mathcal{G} \leq \text{Aut}\mathcal{M}$,

02 Construct $\mathcal{M}/B_G = \langle S', A', \Psi', P', R' \rangle$.

03 Set $Que$ to some initial state $\{s_0\}$, $S' \leftarrow \{s_0\}$

04 While $Que$ is non-empty

05 $s = \text{dequeue}(Que)$

06 For all $a \in A_s$

07 If $(s, a) \notin G (s', a')$ for any $(s', a') \in \Psi'$, then

08 $\Psi' \leftarrow \Psi' \cup (s, a)$

09 $A' \leftarrow A' \cup a$

10 $R'(s, a) = R(s, a)$

11 For all $t \in S$ such that $P(s, a, t) > 0$

12 If $t \equiv G_S s'$, for some $s' \in S'$,

13 $P'(s, a, s') \leftarrow P'(s, a, s') + P(s, a, t)$

14 else

15 $S' \leftarrow S' \cup t$

16 $P'(s, a, t) = P(s, a, t)$

17 add $t$ to $Que$