Abstraction

• A key reason that humans are effective problem solvers
  – Learn and plan at a higher level
  – Knowledge transfer
  – c.f. macros, chunks, skills, behaviors, . . .

• Temporal abstraction or plan abstraction
• Spatial abstraction
• Combination of the two

Motivation

• Well studied problem in AI
• Focus of thesis:
  – Decision theoretic setting
    • Markov decision processes
  – General framework
    • Accommodate different notions of abstraction
      – Aggregation, symmetry (Zinkevich and Balch ’01), Popplestone and Grupen ’00; projections, structured abstractions (Boutilier et al. ’94, ’95, ’01)
    – Formal algebraic framework
      • Group theory, model minimization, operations research
  – Combination of temporal and spatial abstraction
    • Behaviors in a relative frame of reference
      – Efficient knowledge transfer

Outline of Thesis

• Abstraction in decision making
  – Algebraic framework
  – Exploiting symmetry and structure
  – Approximate equivalence
• Abstraction in hierarchical reinforcement learning
  – Hierarchical task decomposition
  – Relativized options
  – Algorithms for dynamic abstraction
    • Choosing transformations
    • Deictic representation

Outline of Thesis

• Abstraction in decision making
  – Algebraic framework
  – Exploiting symmetry and structure
  – Approximate equivalence
• Abstraction in hierarchical reinforcement learning
  – Hierarchical task decomposition
  – Relativized options
  – Algorithms for dynamic abstraction
    • Choosing transformations
    • Deictic representation
Outline of Talk

• Abstraction in decision making
  – Algebraic framework
  – Approximate equivalence
• Abstraction in hierarchical reinforcement learning
  – Relativized options
  – Algorithms for dynamic abstraction
• Choosing transformations

Outline of Talk

• Abstraction in decision making
  – Algebraic framework
  – Approximate equivalence
• Abstraction in hierarchical reinforcement learning
  – Relativized options
  – Algorithms for dynamic abstraction
• Choosing transformations

Markov Decision Processes

• MDP, $M$, is the tuple: $M = \langle S, A, \Psi, P, R \rangle$
  – $S$: set of states.
  – $A$: set of actions.
  – $\Psi \subseteq S \times A$: set of admissible state-action pairs.
  – $P: \Psi \times S \to [0,1]$: probability of transition.
  – $R: \Psi \to \mathbb{R}$: expected reward.
• Policy $\pi: S \to A$ (can be stochastic)
• Maximize total expected reward.

Homomorphisms

Group homomorphism

Let $G$ and $G'$ be groups with operations $+$ and $\cdot'$ respectively
$h: G \to G'$ is a group homomorphism iff
$h(x + y) = h(x) + h(y)$ $\forall x, y \in G$

\[
\begin{array}{ccc}
G \times G & \xrightarrow{+} & G \\
G \times G & \xrightarrow{\cdot'} & G'
\end{array}
\]

Example

Homomorphisms (cont.)

Automaton homomorphism

in the autonomous case:

\[
M = \langle S, \delta \rangle, \quad M' = \langle S', \delta' \rangle
\]

$h(\delta(x)) = \delta'(h(x))$

induces equivalence classes in $S$
MDP Homomorphism

MDPs $M = (S, A, P, R, \Psi)$, $M' = (S', A', P', R', \Psi')$ surjection $h: \Psi \rightarrow \Psi'$ defined by $h(s, a) = (f(s), g, (a))$ where:

1. $f: S \rightarrow S'$, $g_i: A_i \rightarrow A'_i$, for all $i \in S$, are surjections such that for all $s \in S$, and $a_1 \in A$,:
   - $f_i(s) = s'$ then $a' \in A'$ such that $g_i(a) = a'$.
2. $R'(f(s), g_i(a)) = R(s,a)$

\[
\begin{array}{c}
\begin{pmatrix}
(s,a)
\end{pmatrix}
\end{array}
\xrightarrow{P}
\begin{array}{c}
\begin{pmatrix}
(s',a')
\end{pmatrix}
\end{array}
\]

\[
\begin{array}{c}
\begin{pmatrix}
(s,a)
\end{pmatrix}
\end{array}
\xrightarrow{P'}
\begin{array}{c}
\begin{pmatrix}
(s',a')
\end{pmatrix}
\end{array}
\]

\[
\begin{array}{c}
\begin{pmatrix}
(s,a)
\end{pmatrix}
\xrightarrow{R}
\begin{array}{c}
\begin{pmatrix}
(s',a')
\end{pmatrix}
\end{array}
\]

\[
\begin{array}{c}
\begin{pmatrix}
(s,a)
\end{pmatrix}
\xrightarrow{R'}
\begin{array}{c}
\begin{pmatrix}
(s',a')
\end{pmatrix}
\end{array}
\]

\[
\begin{array}{c}
\begin{pmatrix}
(s,a)
\end{pmatrix}
\xrightarrow{h}
\begin{array}{c}
\begin{pmatrix}
(s',a')
\end{pmatrix}
\end{array}
\]

Some Theoretical Results

[generalizing those of Dean and Givan, 1997]

• Optimal Value equivalence:
  If $h(s, a) = (s', a')$ then $Q(s, a) = Q'(s', a')$.

• Corollary:
  If $h(s, a_i) = h(s_i, a_i)$ then $Q'(s_i, a_i) = Q'(s, a_i)$.

Theorem: If $M'$ is a homomorphic image of $M$, then a policy optimal in $M'$ induces an optimal policy in $M$.

• Solve homomorphic image and lift the policy to the original MDP.

Excel Example

Model Minimization

• Finding reduced models that preserve some aspects of the original model

• Various modeling paradigms
  - Finite State Automata (Hartmanis and Stearns ‘66)
  - Transition Behavior
  - Model Checking (Emerson and Sistla ’96, Lee and Yannakakis ’92)
    • Correctness of system models
  - Markov Chains (Kemeny and Snell ’60)
    • Steady state distribution
  - MDPs (Dean and Givan ’97, Ravindran and Barto ’02)
  - Optimal solutions

Outline of Talk

• Abstraction in decision making
  - Algebraic framework
  - Approximate homomorphisms
  - Error bounds

• Abstraction in hierarchical reinforcement learning
  - Relativized options
  - Algorithms for dynamic abstraction
  - Choosing transformations
  - Summary
Approximate Notions of Equivalence

- Complete and exact equivalence often do not exist.
- Approximate equivalence.
  - "Equivalent" state-action pairs have nearly same behavior.

Approximate Homomorphisms

- Use averages
- Relax homomorphism criteria:
  - \( P'(f(s), g(a), f(\tau)) = \sum_{s', a} P(s, a, s') \)
  - Compute \( \sum_{s', a} P(s, a, s') \) for all \((s, a)\)

Error Bound

- Approximate homomorphism between arbitrarily different MDPs!
- Useful when loss in performance is acceptable.
- Bound the maximum difference in optimal value function in \(M\) and the value of the lifted optimal policy.
  - Specializes Whitt ‘78.
  - Function of maximum difference in the probabilities and rewards that are averaged.

Error bound (cont.)

- \( K_\Delta - \) maximum difference between \( P' (f(s), g(a), f(\tau)) \) and \( \sum_{s', a} P(s, a, t) \)
- \( K_r - \) corresponding difference in reward
- \( \Delta - \) the range of the reward function
- \( \gamma - \) the discount factor, \( 0 \leq \gamma < 1 \)

\[
\|V' - V''\|_{\text{max}} \leq \frac{2}{1-\gamma} \left( K_\Delta + \frac{\gamma}{1-\gamma} K_r \right)
\]

Bounded Parameter Approximation

- Model as a map onto a Bounded-parameter MDP (Givan, Leach and Dean ‘00)
  - Transition probabilities and rewards given by bounded intervals
  - Upper and lower bounds on optimal values of states
  - Loose bounds
Outline of Talk

- Abstraction in decision making
  - Algebraic framework
  - Approximate equivalence
- Abstraction in hierarchical reinforcement learning
  - Semi-Markov decision processes
  - Options framework
  - Relativized options
  - Algorithms for dynamic abstraction
  - Choosing transformations
- Summary

Example Revisited

Optimistic Case

Pessimistic Case

(discrete-time) semi-Markov Decision Process

- SMDP, $M$, is the tuple: $M = (S, A, \Psi, P, R)$
  - $S$: set of states.
  - $A$: set of actions.
  - $\Psi \subseteq S \times A$: set of admissible state-action pairs.
  - $P: \Psi \times S \times N \to [0,1]$: transition probabilities.
  - $R: \Psi \times N \to \mathbb{R}$: expected reward.
- Policy (stationary, stochastic): $\pi: \Psi \to [0,1]$
- Maximize expected return.
- Generalize MDP homomorphism.

Hierarchical Reinforcement Learning

Options (Sutton, Precup, & Singh, 1999): A generalization of actions to include temporally-extended courses of action

Example: robot docking

Sub-goal Options

- Task is to collect all objects in the world
- 5 options – one for each room.
- Markov, subgoal options
- Implicitly define option policy
- Employ option specific abstraction

Relativized Options

Relativized option:

$O = (h, M_o, I, \beta)$

$h$: Option homomorphism.

$M_o$: Option SMDP. (Image of $h$)

$I \subseteq S$: Initiation set.

$\beta: S_o \to [0,1]$: Termination criterion.
Rooms world task
- Task is to collect all objects in the world
- 5 options – one for each room
- Single relativized option – get-object-exit-room
- Partial homomorphism
- Especially useful when learning option policy
  - Speed up.
  - Knowledge transfer.

Experimental Setup
- Regular Agent
  - 5 options, one for each room
  - Option reward of +1 on exiting room with object
- Relativized Agent
  - 1 relativized option, known homomorphism
  - Same option reward
- Global reward of +1 on completing task
- Actions fail with probability 0.1

Learning Algorithm
- Hierarchical SMDP Q-learning (Dietterich ’00b)
  - Q-learning at the lowest level (Watkins ’89)
  - SMDP Q-learning at the higher levels (Bradtke and Duff ’95)
- Simultaneous learning at all levels
  - Converges to recursively optimal policy
    - Using results from Dietterich ’00a

Results
- Average over 100 runs

Asymmetric Testbed

Results – Asymmetric Testbed
- Still significant speed up in initial learning
- Asymptotic performance slightly worse
Outline of Talk

- Abstraction in decision making
  - Algebraic framework
  - Approximate equivalence
- Abstraction in hierarchical reinforcement learning
  - Relativized options
  - Algorithms for dynamic abstraction
  - Choosing transformations
  - Summary

Choosing Transformations

Motivation

- Relax prior knowledge requirement
  - Unknown homomorphism
- Option SMDP and policy can be viewed as a policy schema (Schmidl '75, Arbib '95)
  - Template of a policy
  - Acquire schema in a prototypical setting
  - Learn bindings of sensory inputs and actions to schema
- Assume set of possible bindings available

Choosing Transformations

Problem Formulation

- Given:
  - \( M_{O,I,\beta} \) of a relativized option
  - \( H \), a family of transformations
- Identify the option homomorphism \( h \)
- Formulate as a parameter estimation problem
  - One parameter, takes values from \( H \)
  - Samples: \( \{ s_n, a_n, s_{n+1} \} \)
  - Bayesian learning

Algorithm

- Assume uniform prior: \( p_0(h, \bar{x}) \)
- Experience: \( \{ s_n, a_n, s_{n+1} \} \)
- \( P(\{ s_n, a_n, s_{n+1} \} | h, \bar{x}) = P_h(f(s_n), g_\alpha(a_n), f(s_{n+1})) \)
- Update Posteriors:
  \[
  p_n(h, \bar{x}) = \frac{P_h(f(s_n), g_\alpha(a_n), f(s_{n+1}))}{\text{Normalizing Factor}}
  \]

Rooms world task

- Train in room 1
- 8 candidate transformations
  - Reflections about x and y axes and the \( x=y \) and \( x=-y \) lines
  - Rotations by integer multiples of 90 degrees

Results – Speed of convergence

- Not much of a difference since the task is too easy
- Correct transformation identified in 15 iterations
Choosing Transformations

Approximate Equivalence

• More complex domains
• Problem with Bayesian update
  – Use prototypical room as option schema
  – Susceptible to incorrect samples
• Use a heuristic lower bound

Example

\[ p_s(h, \mathbf{x}) = \frac{P_s(f(s), g_x(a), f(s_{s+1})) \cdot w_{s+1}(h, \mathbf{x})}{\text{Normalization Factor}} \]

Choosing Transformations

Heuristic Update Rule

• Use a heuristic update rule:

\[ w_s(h, \mathbf{x}) = P(f(s), g_x(a), f(s_{s+1})) \cdot w_{s+1}(h, \mathbf{x}) \]

where, \( P(s, a, s') = \max(\nu, P_{s+1}(s, a, s')) \)
and \( \nu \) is a small positive constant.

Complex Game World

• Gather all 4 diamonds in the world
• 25 x 10^3 states
• 40 transformations
  – 8 spatial transformations combined with 5 projections

Experimental Setup

• Regular agent
  – 4 sub-goal options
• Relativized agent
  – Uses option MDP shown earlier
  – Chooses from 40 transformations
• Room 2 has no right transformation
• Hierarchical SMDP Q-learning

Results

Speed of Convergence

• Learning the policy is more difficult than learning the correct transformation!
Results
Transformation Weights in Room 4

• Transformation 12 eventually converges to 1

Results
Transformation Weights in Room 2

• Weights oscillate a lot
• Some transformation dominates eventually
  – Changes from one run to another

Choosing Transformations

• Related work
  – Multiple forward models (Harano et al. ’01, Doya et al. ’02)
  – Dynamic control models (Costa and Grupen ’98)
  – Variably bound controllers (Huber and Grupen ’99)
• Representations can be designed to implicitly perform transformations
  – Formalizes such representations
  – E.g. Deictic representations

Summary of Contributions

• Developed an abstraction framework for MDPs
  – Introduced MDP homomorphisms
    • State dependent action recoding
    • Theoretical results
• Approximate homomorphisms
  – Bound maximum loss
  – Upper and lower bound performance

Summary of Contributions (cont.)

• Abstraction in hierarchical systems
  – Relativized options
    • An option defined in a relative frame of reference
    • Uses partial homomorphisms
  – Policy schema
    • Policy template
  – Bayesian algorithm for choosing the right bindings
    • Heuristic modification for approximate equivalence
    • Complex game domain

Other Contributions

• Exploiting structure and symmetry
  – Structured morphisms
  – Symmetry groups
    • Reflections, rotations and permutations
    • Polynomial time algorithm
• Hierarchical decomposition framework
  – Based on SMDP homomorphisms
  – Relation to safe state abstraction (Dietterich ’00a)
• Deictic option schema
  – Representation based on pointers (Agre ’88)
  – Modification of Bayesian algorithm
Future Work

• Practical application of framework
  – Humanoid experiments
• Abstraction algorithms
  – Symbolic representations (Feng et al. ’02, ’03)
• Relation to partial observability
• Relation to other abstract representations
  – Probabilistic relational models (Getoor et al. ’01)