Lazy Approximation
An approach for solving continuous finite-horizon MDPs

Presented by Lihong Li
Joint work with Michael Littman
With thanks to N. Meuleau and other RL$^3$ members

AAAI-05 / Rutgers Lab for Real-Life Reinforcement Learning (RL$^3$)
Introduction

- Markov decision processes (MDPs)
  - Framework for decision-theoretic planning
- Many problems are continuous in nature
  - Transportation scheduling [Boyan & Littman 00]
  - Planetary rover planning [Bresina et al. 02]
Planetary Rover Example

- State components include
  - Remaining energy
  - Remaining execution time

- Continuous state transition
  - E.g., energy consumption of taking a picture

Both continuous

p.d.f. of energy consumption
Solving large-scale MDPs is difficult
- Curse of dimensionality

Solving continuous MDPs is even more difficult
- Need practical representations for
  - models
  - value functions
- Need efficient & stable approximate solutions
Notation

- **MDP**
  - $X$: continuous state space
    - Usually, $X = [0, 1)^k$
  - $A$: finite action set
  - $T$: transition function
  - $R$: reward function

- **Solving MDPs by dynamic programming**
  - Bellman equation

$$V^{n+1}(x) = \max_{a \in A} \left\{ R(x, a) + \int_X T(x'|xa)V^n(x')dx' \right\}$$
Previous Work

- [Boyan & Littman 00]: TiMDP
- [Feng et al. 04]: extension
- Limited to structured MDPs with
  - Reward function
    - PWC or PWLC
    - Continuous
  - Transition function:
    - Discrete
    - Need to pre-specify discretization resolution
    - Referred to as $DM$

Motivation: discretization-free?
LA Summary

Design objectives
- Continuous PWC transitions $T$
- Continuous PWC reward $R$
- Flexible error control
- Flexible function compactness control
  - Compactness: # pieces in PWC functions

Main idea
- Manipulate the continuous model directly
- Postpone approximation until necessary
  - Thus called *lazy approximation* (LA)
  
  But... we’re not lazy
Recall Bellman Equation

\[ V^{n+1}(x) = \max_{a \in A} \left\{ R(x, a) + \int_{x} T(x' | xa) V^n(x') dx' \right\} \]

- Two transition models
  - Absolute model: \( T(x' | xa) = T(x' | a) \)
  - Relative model: \( T(x' | xa) = T(x' - x | a) \)

- Relative model is more challenging
  - Integral becomes a convolution of two PWCs
Basic Idea

\[ V^n \xrightarrow{DP} V^{n+1} \xrightarrow{LA} \tilde{V}^{n+1} \xrightarrow{DP} \ldots \]

**LA:** \[ V^0 \equiv 0 \xrightarrow{DP} \tilde{V}^1 \xrightarrow{DP} \tilde{V}^2 \xrightarrow{LA} \tilde{V}^2 \xrightarrow{DP} \tilde{V}^3 \xrightarrow{LA} \tilde{V}^3 \xrightarrow{DP} \ldots \]

**Small Error:** \[ \epsilon_n = \left\| \tilde{V}^n - \tilde{V}^n \right\|_\infty \]

**Compact Approximation:**

\[ \tilde{V}^n \]

**DM:** \[ V^0 \equiv 0 \xrightarrow{DP} \tilde{W}^1 \xrightarrow{DP} \tilde{W}^2 \xrightarrow{DP} \tilde{W}^3 \xrightarrow{DP} \ldots \]
Extensions

- Multidimensional state spaces
  - PWC: constant in each hyper-rectangle
  - Can use kd-trees [Friedman et al. 77]
  - Convolution of two PWCs

\[ \hat{V}^{n+1}(\vec{x}) = \prod_{i=1}^{k} (a_i x_i + b_i) \]

- Efficient LA:
  - Complexity of finding optimal constant-function approximation within each piece: \( O(k) \)
Extensions (cont’d)

- Non-PWC transition function
  - Approximate it w/ PWC
  - Favored by empirical evidence, over DM

- Dealing w/ discrete state components
  - Rover example
  - No additional essential difficulties
  - Can be handled within the same framework
Error Control

1. Explicitly control the approximation error
2. $L_\infty$-error accumulates additively over horizons

Errors in DM rely on
1. Resolution
2. Smoothness of the value function: not measurable
Compactness Control

1. Explicit tradeoff w/ approximation error
2. Non-uniform partitioning: potentially much more compact at the same error level

Compactness/resolution in DM is usually determined a priori
Experiments

- Randomly generated 1-D problems
  - State space: \([0, 1)\)
  - Horizon: 10
  - PWC reward
  - PWC or Gaussian transitions
    - For Gaussian transitions
      - DM: use discretization
      - LA: use PWC approximation
Horizon = 1
Horizon = 5
Horizon = 10

<table>
<thead>
<tr>
<th></th>
<th>Time (s)</th>
<th>FuncSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>0.008</td>
<td>33</td>
</tr>
<tr>
<td>DM</td>
<td>0.1542</td>
<td>401</td>
</tr>
</tbody>
</table>
Error Results

![Graph showing Gaussian transitions with Max Error on the x-axis and Running Time (seconds) on the y-axis. The graph compares DM, LA (size=20), and LA (epsilon=0.01).]
Compactness Results

Gaussian transitions

- **DM**
- **LA (size=20)**
- **LA (epsilon=0.01)**
One Application

- Planetary rover planning
  - Two-location, two-object problem

- More realistic experiments in the future
Future Work

- Efficient data structures and algorithms for manipulating models
  - computing convolution is quite expensive with kd-trees
  - lazy approximation with specified error level
    - currently, greedy alg for high dimensional space

- Relax some structural constraints

- Implement wait/dawdle actions
  - [Boyan & Littman 00]
  - E.g., wait until sunrise to take high-quality pictures

- Real-world applications
  - E.g., planetary rover planning
Conclusions

- Developed *Lazy Approximation*
  - Solving continuous structured MDPs
  - Discretization-free
  - Flexible error control
  - Flexible compactness control

- For more details:
  - Rutgers CS Tech Report #577
  - RL³: [http://www.cs.rutgers.edu/rl3](http://www.cs.rutgers.edu/rl3)

- Questions & comments?