Thresholded Rewards: Acting Optimally in Timed, Zero-Sum Games

Colin McMillen and Manuela Veloso

Presenter: Man Wang
Overview

- Zero-sum Games
- Markov Decision Problems
- Value Iteration Algorithm
- Thresholded Rewards MDP
- TRMDP Conversion
- Solution Extraction
- Heuristic Techniques
- Conclusion
- References
Zero-sum Games

Zero–sum game
A participant's gains of utility -- Losses of the other participant

Cumulative intermediate reward
The difference between our score and opponent’s score

True reward
Win, loss or tie
Determined at the end based on intermediate reward
Markov Decision Problem

- Consider a non-perfect system
- Actions are performed with a probability less than 1
- What is the best action for an agent under this constraint?
- Example: A mobile robot does not exactly perform the desired action
Markov Decision Problem

- Sound means of achieving optimal rewards in uncertain domains
- Find a policy maps state $S$ to action $A$
- Maximize the cumulative long-term rewards
Value Iteration Algorithm

What is the best way to move to +1 without moving into -1?

Consider non-deterministic transition model:
Calculate the utility of the center cell:

\[
U_{t+1}(i) = R(i) + \max_a \sum_j M_{ij}^a \cdot U_t(j)
\]

\[
= \text{reward} + \max\{ \\
0.1 \cdot 1 + 0.8 \cdot 5 + 0.1 \cdot 10 \quad (\leftarrow), \\
0.1 \cdot 5 + 0.8 \cdot 10 + 0.1 \cdot -8 \quad (\uparrow), \\
0.1 \cdot 10 + 0.8 \cdot -8 + 0.1 \cdot 1 \quad (\rightarrow), \\
0.1 \cdot -8 + 0.8 \cdot 1 + 0.1 \cdot 5 \quad (\downarrow)\}
\]

\[
= 1 + \max\{5.1 (\leftarrow), 7.7 (\uparrow), \\
-5.3 (\rightarrow), 0.5 (\downarrow)\}
\]

\[
= 1 + 7.7
\]

\[
= 8.7
\]
Value Iteration Algorithm

1. The given environment.

2. Calculate Utilities.

3. Extract optimal policy.

4. Execute actions.
Thresholded Rewards MDP

\[ \text{TRMDP} \left( M, f, h \right): \]

- **\( M \): MDP(\( S, A, T, R, s_0 \))**
- **\( f \): threshold function**
  \[ f(r_{\text{intermediate}}) = r_{\text{true}} \]
- **\( h \): time horizon**

\[
r_{\text{true}} = \begin{cases} 
1 & \text{if } r_{\text{intermediate}} > 0 \\
0 & \text{if } r_{\text{intermediate}} = 0 \\
-1 & \text{if } r_{\text{intermediate}} < 0.
\end{cases}
\]

**Algorithm 1** Dynamics of a thresholded-rewards MDP.

\[
s \leftarrow s_0 \\
r_{\text{intermediate}} \leftarrow 0 \\
\text{for } t \leftarrow h \text{ to } 1 \text{ do} \\
\quad a \leftarrow \pi(s, t, r_{\text{intermediate}}) \\
\quad s \leftarrow T(s, a) \\
\quad r_{\text{intermediate}} \leftarrow r_{\text{intermediate}} + R(s) \\
\quad r_{\text{true}} \leftarrow f(r_{\text{intermediate}})
\]

Example:

- **States:**
  1. **FOR**: our team scored (reward +1)
  2. **AGAINST**: opponent scored (reward -1)
  3. **NONE**: no score occurs (reward 0)

- **Actions:**
  1. Balanced
  2. Offensive
  3. Defensive
Thresholded Rewards MDP

**Expected one step reward:**

1. **Balanced:** $0 = 0.05 \times 1 + 0.05 \times (-1) + 0.9 \times 0$

2. **Offensive:** $-0.25 = 0.25 \times 1 + 0.5 \times (-1) + 0.25 \times 0$

3. **Defensive:** $-0.01 = 0.01 \times 1 + 0.02 \times (-1) + 0.97 \times 0$

Suboptimal solution, true reward = 0
TRMDP Conversion
Algorithm 2 Converts a TRMDP \((M, f, h)\) into an MDP \(M'\) suitable for finding the optimal thresholded-rewards policy.

1: **Given:** MDP \(M = (S, A, T, R, s_0)\), threshold function \(f\), time horizon \(h\)

2: \(s'_0 \leftarrow (s_0, h, 0)\)

3: \(S' \leftarrow \{s'_0\}\)

4: **for** \(i \leftarrow h\) to 1 **do**

5: **for all** states \(s'_1 = (s_1, t, ir) \in S'\) such that \(t = i\) **do**

6: **for all** transitions \(T(s_1, a, s_2)\) in \(M\) **do**

7: \(s'_2 \leftarrow (s_2, t - 1, ir + R(s_2))\)

8: \(S' \leftarrow S' \cup \{s'_2\}\)

9: \(T'(s'_1, a, s'_2) = T(s_1, a, s_2)\)

10: **for all** states \(s' = (s, t, ir)\) in \(M'\) **do**

11: **if** \(t = 0\) **then**

12: \(R'(s') \leftarrow f(ir)\)

13: **else**

14: \(R'(s') \leftarrow 0\)

15: **return** \(M' = (S', A, T', R', s'_0)\)
TRMDP Conversion

The MDP $M'$ given MDP $M$ and $h=3$
Two important facts:

- $M'$ has a layered, feed-forward structure: every layer contains transitions only into the next layer.
- At iteration $k$ of value iteration, the only values that change are those for the states $s'=(s, t, ir)$ such that $t=k$. 
Expected reward = 0.1457

Win : 50%
Lose: 35%
Tie : 15%

Optimal policy for M and h=120
Solution Extraction

Effect of changing opponent’s capabilities

Performance of MER vs TR on 5000 random MDPs
Heuristic Techniques

- Uniform-k heuristic
- Lazy-k heuristic
- Logarithmic-k-m heuristic
- Experiments
Uniform-k heuristic

- Adopt non-stationary policy
- Change policy every k time steps
- Compress the time horizon uniformly by factor k
- Solution is suboptimal
Lazy-k heuristic

- More than k steps remaining:
  No reward threshold

- K steps remaining:
  Create threshold rewards MDP
  Time horizon k
  Current state as initial state
Logarithmic-k-m heuristic

- Time resolution becomes finer when approaching the time horizon
- $k$ – Number of decisions made before the time resolution increased
- $m$ – The multiple by which the resolution is increased
- For instance, $k=10, m=2$ means that 10 actions before each increase, time resolution doubles on each increase
Experiment

60 different MDPs randomly chosen from the 5000 MDPs in previous experiment

Uniform-k suffers from large state size
Logarithmic highly depend on parameters
Lazy-k provides high true reward with low number of states
Conclusion

- Introduced thresholded-rewards problem in finite-horizon environment
  - Intermediate rewards
  - True reward at the end of horizon
  - Maximize the probability of winning
- Present an algorithm converts base MDP to expanded MDP
- Investigate three heuristic techniques generating approximate solutions
References


