Potential Based Reward Shaping Tutorial

ALA 2014
Sam Devlin
Knowledge-Based Reinforcement Learning

- Commonly, RL algorithms assume no prior knowledge
- Including domain knowledge can simplify learning
Reward Shaping

Q-Learning

- A popular RL algorithm
- \( Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q_i(s', a') - Q(s, a)] \)

Reward Shaping

- Provide heuristic knowledge by an additional reward
- \( Q(s, a) \leftarrow Q(s, a) + \alpha [r + F(s, s') + \gamma \max_{a'} Q_i(s', a') - Q(s, a)] \)
Potential-Based Reward Shaping Tutorial

Introduction

Potential-Based Reward Shaping

$F(s, s') = \gamma \Phi(s') - \Phi(s)$

- $F(s, s')$: Additional reward when moving from state $s$ to $s'$
- $\gamma$: Discount factor
- $\Phi(s)$: Potential of state $s$
Potential-Based Reward Shaping Tutorial

Potential-Based Reward Shaping

Formal Definition

- $F(s, s') = \gamma \Phi(s') - \Phi(s)$

Guarantees

- Policy invariance (optimal policy unchanged) in single agent\(^1\)

Can

- Increase/Decrease time taken to learn optimal policy

---

\(^1\) Ng, Russell and Harada. “Policy Invariance Under Reward Transformations: Theory And Application To Reward Shaping.” ICML, 1999.
An Example Potential Function
Potential-Based Reward Shaping

Figure: A Typical Single Agent Result
Proof of Policy Invariance

\[ U_\Phi(\bar{s}) = \sum_{j=0}^{\infty} \gamma^j (r_j + \gamma \Phi(s_{j+1}) - \Phi(s_j)) \]

\[ = \sum_{j=0}^{\infty} \gamma^j r_j + \sum_{j=0}^{\infty} \gamma^{j+1} \Phi(s_{j+1}) - \sum_{j=0}^{\infty} \gamma^j \Phi(s_j) \]

\[ = U(\bar{s}) + \sum_{j=1}^{\infty} \gamma^j \Phi(s_j) - \Phi(s_0) - \sum_{j=1}^{\infty} \gamma^j \Phi(s_j) \]

\[ = U(\bar{s}) - \Phi(s_0) \]

---

Proof of Policy Invariance

\[ U_\Phi(\bar{s}) = \sum_{j=0}^{\infty} \gamma^j (r_j + \gamma \Phi(s_{j+1}) - \Phi(s_j)) \]
Proof of Policy Invariance

\[ U_\Phi(\bar{s}) = \sum_{j=0}^{\infty} \gamma^j (r_j + \gamma \Phi(s_{j+1}) - \Phi(s_j)) \]

\[ = \sum_{j=0}^{\infty} \gamma^j r_j + \sum_{j=0}^{\infty} \gamma^{j+1} \Phi(s_{j+1}) - \sum_{j=0}^{\infty} \gamma^j \Phi(s_j) \]
Potential Based Reward Shaping Tutorial

Proof of Policy Invariance

$$U_\Phi(\vec{s}) = \sum_{j=0}^{\infty} \gamma^j (r_j + \gamma \Phi(s_{j+1}) - \Phi(s_j))$$

$$= \sum_{j=0}^{\infty} \gamma^j r_j + \sum_{j=0}^{\infty} \gamma^{j+1} \Phi(s_{j+1}) - \sum_{j=0}^{\infty} \gamma^j \Phi(s_j)$$

$$= U(\vec{s}) + \sum_{j=1}^{\infty} \gamma^j \Phi(s_j) - \Phi(s_0) - \sum_{j=1}^{\infty} \gamma^j \Phi(s_j)$$
Proof of Policy Invariance

\[ U_{\Phi}(\bar{s}) = \sum_{j=0}^{\infty} \gamma^j (r_j + \gamma \Phi(s_{j+1}) - \Phi(s_j)) \]

\[ = \sum_{j=0}^{\infty} \gamma^j r_j + \sum_{j=0}^{\infty} \gamma^{j+1} \Phi(s_{j+1}) - \sum_{j=0}^{\infty} \gamma^j \Phi(s_j) \]

\[ = U(\bar{s}) + \sum_{j=1}^{\infty} \gamma^j \Phi(s_j) - \Phi(s_0) - \sum_{j=1}^{\infty} \gamma^j \Phi(s_j) \]

\[ = U(\bar{s}) - \Phi(s_0) \]
Q-Table Initialization

▶ Wiewiora: “Potential-based shaping and Q-value initialization are equivalent.” (JAIR, 2003)

▶ ...If the potential function is static.
Multi-Agent Reinforcement Learning

- Multiple agents learning concurrently in the same environment
- Typically learn a Nash equilibrium
- No clear notion of an optimal policy
Multi-Agent Potential-Based Reward Shaping

Guarantees

- Nash Equilibria not altered

Can

- Increase/Decrease time taken to reach a stable joint policy
- Change final joint policy

---

Potential-Based Reward Shaping

**Figure:** A Typical Multi-Agent Result
Exploration Altered

- Reward shaping alters exploration / which actions are chosen
- In single-agent, this affects time to convergence
- In multi-agent, this may cause the agents to reach a different point of equilibrium
  - Wellman and Hu (1998) showed the joint policy converged upon in a learning MAS is highly sensitive to initial belief
Multi-Agent Example 4

Results

(a) Without Reward Shaping
Safe Reward Shaping
Safe Reward Shaping

Diagram:
- Start at 5
- Movements and rewards:
  - From 5 to -5: +5
  - From 5 to 10: +5
  - From 5 to 15: +5
  - From -5 to 0: -5
  - From 0 to 10: +5
  - From 0 to 15: +5
  - From 0 to 5: 0
  - From 0 to 0: 0
  - From 0 to -10: -10
  - From 10 to -5: -10
  - From 15 to -5: -10
Results

(b) With Safe Reward Shaping
Coordinated Reward Shaping
Results

(c) With Coordinated Reward Shaping
Miscoordinated Reward Shaping
Results

(d) With Miscoordinated Reward Shaping
Multiagent Example 2: RoboCup KeepAway

---

Practical Exercise: Design a Potential Function

Figure: State Representation

- a: $\text{dist}(K_1, K_2)$
- b: $\text{dist}(K_1, K_3)$
- c: $\text{dist}(K_1, T_1)$
- d: $\text{dist}(K_1, T_2)$
- e: $\text{dist}(K_1, C)$
- f: $\text{dist}(K_2, C)$
- g: $\text{dist}(K_3, C)$
- h: $\text{dist}(T_1, C)$
- i: $\text{dist}(T_2, C)$
- j: $\text{min}_\text{dist}(K_{2\text{-mid}}, T_j)$
- k: $\text{min}_\text{dist}(K_{3\text{-mid}}, T_j)$
- l: $\text{min}_\text{ang}(K_2, K_1, T_j)$
- m: $\text{min}_\text{ang}(K_3, K_1, T_j)$
Past Assumptions

- Previous theoretical guarantees assume a static potential function

- Some claim the potential function must converge before the agent can

---

Dynamic Potential Based Reward Shaping

Finally, Guarantees policy invariance or consistent Nash equilibria, provided:

\[ F(s, t, s', t') = \gamma \Phi(s', t') - \Phi(s, t) \]

---

Context Sensitive Reward Shaping

In different contexts we often recommend different behaviours

---

Belief Revision

Q-Table Initialization

- Wiewiora: “Potential-based shaping and Q-value initialization are equivalent.” (JAIR, 2003)

- ...If the potential function is static.
State and Action Shaping \(^{10}\)

**Look-Ahead Advice**

- \( F(s, a, s', a') = \gamma \Phi(s', a') - \Phi(s, a) \)
- \( \pi(s) = \arg\max_a \{ Q(s, a) + \Phi(s, a) \} \)
- Maintains all previous guarantees

**Look-Back Advice**

- \( F(s, a, s', a') = \Phi(s', a') - \gamma^{-1} \Phi(s, a) \)
- No guarantees proven

Partial Observability

Formal Definition

\[ F(o, o') = \gamma \Phi(o') - \Phi(o) \]

Guarantees

- Equivalence to Q-table initialisation
- Policy invariance (optimal policy unchanged) in single agent
- Consistant Nash equilibria in multi-agents systems

---

\[ ^{11} \text{Eck, Soh, Devlin and Kudenko. “Potential-Based Reward Shaping for Partially Observable Markov Decision Processes.” AAMAS, 2013.} \]
Closing Remarks
Implementation Advice

- $\gamma$ must be equal to update rule
- Use an absorbing state
- Store current potential for next iteration
- Avoid negative potentials \(^{12}\)

General Effect

- Does not modify any property of the underlying MDP or SG invariant to changes in absolute value of expected return.

- Provided a property is only reliant on the relative difference or order of expected returns, potential-based reward shaping will not affect it.
Neccessity

- For every reward shaping function that is not potential-based, there is an MDP where the optimal policy differs with and without reward shaping.  

References


- Wiewiora. “Potential-based shaping and Q-value initialization are equivalent.” JAIR, 2003


Potential-Based Reward Shaping Tutorial

Closing Remarks

AAMAS 2014

Potential-Based Difference Rewards for Multiagent Reinforcement Learning.

Sam Devlin, Logan Yliniemi, Daniel Kudenko and Kagan Tumer

Learning I
Miles Davis A & B
Wednesday 09:00