Behavior Clustering Inverse Reinforcement Learning and Approximate Optimal Control with Temporal Logic Tasks

By
Siddharthan Rajasekaran

Committee:
Jie Fu (Advisor), Jane Li (Co-advisor), Carlo Pinciroli
Outline

- Part I
  - Behavior Clustering Inverse Reinforcement Learning
- Part II
  - Approximate Optimal Control with Temporal Logic Tasks
Part I

Behavior Clustering Inverse Reinforcement Learning
Outline

Behavior Clustering Inverse Reinforcement Learning

- Learning from demonstrations
  - Related work
    - Behavior cloning
    - Reinforcement Learning
- Background
  - Feature Expectation
  - Maximum Entropy IRL
- Motivation
- Method
- Results
- Conclusion
- Future work
Related work - Broad Overview

- Learning from demonstrations
  - Behavioral cloning
  - Reward shaping towards demonstrations
  - Inverse Reinforcement Learning
Related work - Broad Overview

- Learning from demonstrations
  - Behavioral cloning - Bojarski et. al. 2016, Ross et. al. 2011
    - Treat demonstrations as labels and perform supervised learning
    - Simple to use and implement
    - Does not generalize well
    - Crash due to positive feedback
  - Given trajectories
    \[ \tau = \{(s_0, a_0), (s_1, a_1), (s_2, a_2), \cdots \} \]
    - Learn the function approximation \( f : S \rightarrow A \)
Related work - Broad Overview

- Learning from demonstrations
  - Reward shaping towards demonstrations - Brys et. al. 2015, Vasan et. al. 2017
    - Give auxiliary reward for mimicking the expert
    - Does not generalize well
    - Requires definition of distance metrics

Diagram:
- Demonstration
- Closest point
- Mimicking action
Related work

- Learning from demonstrations
  - Inverse Reinforcement Learning - Abbeel et. al. 2004, Ziebart et. al. 2008
    - Finds the reward the expert is maximizing
    - Generalizes well to unseen situations

- This is the topic of interest
Related work

- Why Inverse Reinforcement Learning?
  - Finding the intent
    - Useful for reasoning the decisions of expert
  - Prediction
    - Plan ahead of time
  - Collaboration
    - Assist humans to complete a task
IRL Motivation

- Motivating example
IRL Motivation

- An autonomous agent practicing IRL
  - Recognize intent
  - Take actions completely different from the expert to serve the intent

IRL practitioner

Warneken & Tomasello 2006
An autonomous agent practicing IRL
- Recognize intent
- Take actions completely different from the expert to serve the intent

Warneken & Tomasello 2006
Preliminaries
Agent interaction modeling in Reinforcement Learning

Uses policy $\pi : S \rightarrow A$

$s_t \in S$

$a_t \in A$
Agent interaction modeling in Reinforcement Learning

Uses policy $\pi : S \rightarrow A$

$s_t \in S$

$a_t \in A$

$\gamma \in [0, 1)$

Objective of RL: $\max_{\pi} r_1 + \gamma r_2 + \gamma^2 r_3 \cdots$
Reinforcement Learning

- Given
  - Environment
  - Set of actions to choose from
  - Rewards
- Finds
  - The optimal behavior to maximize cumulative reward
Reinforcement Learning
- Given
  - Environment
  - Set of actions to choose from
  - Rewards
- Finds
  - The optimal behavior to maximize cumulative reward
Inverse Reinforcement Learning

- Given
  - Environment
  - Set of actions to choose from
  - Expert demonstrations

- Finds
  - The best reward function that explains the expert demonstrations
We will introduce
  - Linear Reward Setting
  - Feature expectation
  - Graphical interpretation of RL
    • Required for graphical interpretation of IRL
• Linear Rewards
  – Linear only in weights: \( r(s) = w^T \phi(s) \)
  – Can be complex and nonlinear in states
    • Using non-linear features
Linear reward - simple example

- Grid world
  - Each color is a region

\[
\phi(s) = \begin{bmatrix}
  1 \\
  1 \\
  1 \\
  1 \\
\end{bmatrix}
\]

Where $I_1$ is the indicator feature if the agent is in red region, $I_2$ if it is in orange region and so on...
Linear reward - simple example

- Grid world
  - Each color is a region

- Reward function
  - Red = +5
  - Yellow = -1

Where $I_1$ is the indicator feature if the agent is in red region, $I_2$ if it is in orange region and so on ...
Linear reward - simple example

- Grid world
  - Each color is a region

- Reward function
  - Red = +5
  - Yellow = -1

\[ r(s) = w^T \phi(s) \]

where, \( w_1 = +5 \) and \( w_3 = -1 \)

- Each dimension in the feature vector is an indicator if we are in that region

Where \( I_1 \) is the indicator feature if the agent is in red region, \( I_2 \) if it is in orange region and so on ...
RL Objective:

\[
\max_{\pi} E \left[ r(s_0) + \gamma r(s_1) + \gamma^2 r(s_2) + \cdots \right]
\]

Where \( l_1 \) is the indicator feature if the agent is in red region, \( l_2 \) if it is in orange region and so on ...
RL Objective:

\[
\begin{align*}
\max_{\pi} & \ E \left[ r(s_0) + \gamma r(s_1) + \gamma^2 r(s_2) + \cdots \right] \\
\max_{\pi} & \ E \left[ w^T \phi(s_0) + \gamma w^T \phi(s_1) + \gamma^2 w^T \phi(s_2) + \cdots \right] \\
\max_{\pi} & \ w^T E \left[ \phi(s_0) + \gamma \phi(s_1) + \gamma^2 \phi(s_2) + \cdots \right] \\
\max_{\pi} & \ w^T \mu(\pi)
\end{align*}
\]

where, \( \mu(\pi) \) is called the feature expectation.
Feature expectation of any behavior is a vector in n-dimensional space
RL - Linear Setting

- Feature expectation of any behavior is a vector in n-dimensional space
RL Geometrically,

**Objective:** $\max_{\pi} w^T \mu(\pi)$
RL Geometrically,

**Objective:** \( \max_{\pi} w^T \mu(\pi) \)

**Objective:** minimize \( \Psi \)
IRL - Overview

- IRL algorithms

1. Parametrize the reward function as $r(s, a, s'; w)$
2. Initialize $w$ randomly
3. Solve the forward problem (reinforcement learning) problem for the current reward parameters to get the current optimal policy $\pi^o$
4. Compute the difference $e$ between the expert’s policy $\pi^e$ and the current optimal policy $\pi^o$
5. If $e$ is smaller than a threshold, stop. Or else update $w$ in a direction that reduces $e$ then go to step 3
Maximum Entropy Inverse Reinforcement Learning
MaxEnt IRL

- Maximum Entropy IRL (Ziebart 2010)

\[
P(\tau_i) = \frac{e^{w^T \mu(\tau_i)}}{\sum_{\tau} e^{w^T \mu(\tau)}}
\]

\[
\mu(\pi) = E_\pi[\mu(\tau)]
\]
MaxEnt IRL

- Maximum Entropy IRL (Ziebart 2010)

\[ P(\tau_i) = \frac{e^{w^T \mu(\tau_i)}}{\sum_\tau e^{w^T \mu(\tau)}} \]

The MaxEnt policy \( \pi_m \) can be found using soft-value iteration.
MaxEnt IRL

- Maximum Entropy IRL (Ziebart 2010)

\[ P(\tau_i) = \frac{e^{w^T \mu(\tau_i)}}{\sum_{\tau} e^{w^T \mu(\tau)}} \]

The MaxEnt policy \( \pi_m \) can be found using soft-value iteration.

Given the parameters \( w \), soft-value iteration finds \( \pi_m \) to induce \( P(\cdot) \)
MaxEnt IRL

- Maximum Entropy IRL (Ziebart 2010)
  - Objective function given demonstrations $\mathcal{D}$

$$\max_w P(\mathcal{D} | w)$$

$$P(\tau_i) = \frac{e^{w^T \mu(\tau_i)}}{\sum_{\tau} e^{w^T \mu(\tau)}}$$

$$\mathcal{D} \triangleq \{\tau_1, \tau_2, \ldots, \tau_n\}$$
**IRL - Linear setting**

- Gradient ascent on likelihood

\[
\mathcal{L}(w) = \ln P(\mathcal{D}|w)
\]

\[
\nabla_w \mathcal{L}(w) = \mu(\mathcal{D}) - \mu(\pi_w)
\]

\[
w := w + \alpha (\mu(\mathcal{D}) - \mu(\pi_w))
\]

\[
P(\tau_i) = \frac{e^{w^T \mu(\tau_i)}}{\sum_{\tau} e^{w^T \mu(\tau)}}
\]

\[
\mathcal{D} \triangleq \{\tau_1, \tau_2, \ldots, \tau_n\}
\]
IRL - Linear setting

- **MaxEnt IRL Algorithm**
  - Initialize random reward parameter $w$
  - Perform soft-value iteration (RL) to get $\pi_w$
  - Compute $\mu(\pi_w)$
  - Perform the update $w := w + \alpha(\mu(D) - \mu(\pi_w))$

$$P(\tau_i) = \frac{e^{w^T \mu(\tau_i)}}{\sum_{\tau} e^{w^T \mu(\tau)}}$$

$D \triangleq \{\tau_1, \tau_2, \ldots, \tau_n\}$
IRL - Linear setting

Initialize random reward parameter $w$
- Perform soft-value iteration (RL) to get $\pi_w$
- Compute $\mu(\pi_w)$
- Perform the update $w := w + \alpha(\mu(D) - \mu(\pi_w))$

\[
\mathcal{L}(w) = \ln P(D|w) \\
\nabla_w \mathcal{L}(w) = \mu(D) - \mu(\pi_w)
\]
Initialize random reward parameter $w$
- Perform soft-value iteration (RL) to get $\pi_w$
- Compute $\mu(\pi_w)$
- Perform the update $w := w + \alpha(\mu(\mathcal{D}) - \mu(\pi_w))$

$$\mathcal{L}(w) = \ln P(\mathcal{D}|w)$$
$$\nabla_w \mathcal{L}(w) = \mu(\mathcal{D}) - \mu(\pi_w)$$
IRL - Linear setting

Initialize random reward parameter $w$

- Perform soft-value iteration (RL) to get $\pi_w$
- Compute $\mu(\pi_w)$
- Perform the update $w := w + \alpha(\mu(D) - \mu(\pi_w))$

$$\mathcal{L}(w) = \ln P(D|w)$$
$$\nabla_w \mathcal{L}(w) = \mu(D) - \mu(\pi_w)$$
Initialize random reward parameter $w$
- Perform soft-value iteration (RL) to get $\pi_w$
- Compute $\mu(\pi_w)$
- Perform the update $w := w + \alpha(\mu(D) - \mu(\pi_w))$

$$L(w) = \ln P(D|w)$$
$$\nabla_w L(w) = \mu(D) - \mu(\pi_w)$$
IRL - Linear setting

Initialize random reward parameter $w$

Perform soft-value iteration (RL) to get $\pi_w$

Compute $\mu(\pi_w)$

Perform the update $w := w + \alpha(\mu(D) - \mu(\pi_w))$

\[
\mathcal{L}(w) = \ln P(D|w)
\]

\[
\nabla_w \mathcal{L}(w) = \mu(D) - \mu(\pi_w)
\]
IRL - Linear setting

Initialize random reward parameter $w$

- Perform soft-value iteration (RL) to get $\pi_w$
- Compute $\mu(\pi_w)$
- Perform the update $w := w + \alpha(\mu(D) - \mu(\pi_w))$

Define:

$$L(w) = \ln P(D|w)$$

$$\nabla_w L(w) = \mu(D) - \mu(\pi_w)$$
Initialize random reward parameter \( w \)

- Perform soft-value iteration (RL) to get \( \pi_w \)
- Compute \( \mu(\pi_w) \)
- Perform the update \( w := w + \alpha(\mu(D) - \mu(\pi_w)) \)

\[
\mathcal{L}(w) = \ln P(D|w)
\]

\[
\nabla_w \mathcal{L}(w) = \mu(D) - \mu(\pi_w)
\]
IRL - Linear setting

- Problems with MaxEnt
  - Interprets variance in demonstrations as suboptimality
- Problems with MaxEnt
  - Interprets variance in demonstrations as suboptimality
- Consider these demonstrations
IRL - Linear setting

- Problems with MaxEnt
  - Interprets variance in demonstrations as suboptimality
- Consider these demonstrations
- Problems with MaxEnt
  - Interprets variance in demonstrations as suboptimality
- Consider these demonstrations
Why should we not learn the “mean” behavior
   – Wrong prediction
     • Agent now predicts the “mean” behavior
   – Learn the unintended behavior
     • Might learn unsafe behavior (think in case of driving)
   – Wrong intent learned.
     • Cannot collaborate
   – Not practical to get consistent demonstrations
Behavior Clustering IRL

- Behavior Clustering IRL
  - Parametric
    - Clusters/behaviors: $\{c_1, c_2, \cdots c_m\}$
    - Soft clustering, learns: $P(c_j | \tau^i)$
      - Probability that a given demonstration $\tau^i$ belongs to a class $c_j$
    - Learns reward parameters: $\{w_1, w_2, \cdots w_m\}$
  - Non-parametric
    - In addition learns the number of clusters: $m$
Behavior Clustering IRL

- Expectation Maximization
  - Missing data: distribution over behaviors
  - Given data: Demonstrations
- Easier to optimize $\max_{\Theta} \ln P(c, D|\Theta)$ than $\max_{\Theta} \ln P(D|\Theta)$
Behavior Clustering IRL

- The new objective function

\[
\max_{\Theta, \Psi} \sum_{i=1}^{n} \sum_{j=1}^{m} P(c_j | \tau^i, \Theta, \Psi) \ln(P(c_j, \tau^i | \Theta, \Psi))
\]

where,

- \(c_j\) – \(j^{th}\) cluster variable
- \(\tau_i\) – \(i^{th}\) demonstration
- \(\Theta\) – parameters of reward functions of all the clusters
- \(\Psi\) – parameters of the class prior, i.e., \(\Psi(c_j) \triangleq P(c_j)\)

Previous Objective function

For a single behavior:

\[
\max_{w} \sum_{i=1}^{n} \ln(P(\tau^i | w))
\]

For multiple behaviors:

\[
\max_{\Theta} \sum_{i=1}^{n} \ln(P(\tau^i | \Theta))
\]

\(\Theta \triangleq \{w_1, w_2, \cdots w_m\}\)
The new objective function

\[ \mathcal{L}(\Theta, \Psi) = \max_{\Theta, \Psi} \sum_{i=1}^{n} \sum_{j=1}^{m} P(c_j|\tau^i, \Theta, \Psi) \ln(P(c_j, \tau^i|\Theta, \Psi)) \]

Update reward functions using

\[ \nabla_{w_j} \mathcal{L}(\Theta, \Psi) = \sum_{i} \beta_{ij} \left( \mu(\tau^i) - \mu(\pi_j) \right) \]

where, \( \beta_{ij} = P(c_j|\tau^i) \) is the probability that \( i^{th} \) demonstration comes from \( j^{th} \) behavior

\( c_j \) - \( j^{th} \) cluster variable
\( \tau_i \) - \( i^{th} \) demonstration
\( \Theta \) - parameters of reward functions of all the clusters
\( \Psi \) - parameters of the class prior, i.e., \( \Psi(c_j) \triangleq P(c_j) \)
\( \mu(\tau^i) \) - feature expectation of the trajectory \( \tau^i \)
\( \mu(\pi_j) \) - MaxEnt policy using \( w^j \) with start states as in \( \tau^i \)

Update in vanilla MaxEnt IRL:

\[ \nabla_{w_j} \mathcal{L}(\Theta, \Psi) = \sum_{i} \left( \mu(\tau^i) - \mu(\pi_j) \right) \]
Behavior Clustering IRL

- The new objective function

\[ \mathcal{L}(\Theta, \Psi) = \max_{\Theta,\Psi} \sum_{i=1}^{n} \sum_{j=1}^{m} P(c_j | \tau^i, \Theta, \Psi) \ln(P(c_j, \tau^i | \Theta, \Psi)) \]

- Update reward functions using

\[ \nabla_{w_j} \mathcal{L}(\Theta, \Psi) = \sum_i \beta_{ij} \left( \mu(\tau^i) - \mu(\pi_j) \right) \]

- Update the priors using \( \nabla_{\Psi} \mathcal{L}(\Theta, \Psi) = 0 \)

\[ \Psi(c_j) = \frac{\sum_i \beta_{ij}}{n} \]

where, \( \beta_{ij} = P(c_j | \tau^i) \) is the probability that \( i^{th} \) demonstration comes from \( j^{th} \) behavior

- Update in vanilla MaxEnt IRL:

\[ \nabla_{w_j} \mathcal{L}(\Theta, \Psi) = \sum_i \left( \mu(\tau^i) - \mu(\pi_j) \right) \]
Non-parametric Behavior Clustering IRL

- Non-parametric BCIRL
  - Learns the number of clusters
  - We should learn the minimum number of clusters
- Chinese Restaurant Process (CRP) is used for non-parametric clustering
Non-parametric Behavior Clustering IRL

- Chinese Restaurant Process (CRP) is used for non-parametric clustering

\[
x_N | x_1, x_2, \ldots, x_{N-1} = \begin{cases} 
  x_i & \text{with probability } \frac{\text{counts}(x_i)}{n-1+\alpha} \\
  \text{new draw from } G_0 & \text{with probability } \frac{\alpha}{n-1+\alpha}
\end{cases}
\]

Source: Internet (CS224n NLP course, Stanford)
Non-parametric Behavior Clustering IRL

- Non-parametric BCIRL
  - Learns the number of clusters
  - We should learn the minimum number of clusters
- Chinese Restaurant Process (CRP)

\[
x_N | x_1, x_2, \ldots, x_{N-1} = \begin{cases} 
  x_i & \text{with probability } \frac{\text{counts}(x_i)}{n-1+\alpha} \\
  \text{new draw from } G_0 & \text{with probability } \frac{\alpha}{n-1+\alpha}
\end{cases}
\]

- For our problem, we count the soft cluster assignment \( \beta_{ij} \) (probability mass)

\[
c_j | c_1, c_2, \ldots, c_n = \begin{cases} 
  c_k & \text{with probability } \frac{\sum_i \beta_{ik}}{\sum_i \sum_j \beta_{ij} + \alpha} = \frac{\sum_i \beta_{ik}}{n+\alpha} \\
  \text{new draw from } G_0 & \text{with probability } \frac{\alpha}{n+\alpha}
\end{cases}
\]
Algorithm 4 non-parametric BCIRL($\mathcal{D}, \alpha, RL$)

1: $n \leftarrow$ Number of demonstrations in $\mathcal{D}$
2: $nc \leftarrow \emptyset$ ▶ Number of demonstrations in each cluster
3: $P \leftarrow$ The transition probability of the MDP
4: $\beta_i \leftarrow 0 \quad \forall i = 1 : n$
5: $ss \leftarrow$ startStates($\mathcal{D}$)
6: while True do
7:     $m \leftarrow \text{len}(nc)$
8:     $\theta_{(m+1)} \leftarrow$ Random Sample from prior
9:     $p \leftarrow \text{normalize}(\text{merge}(nc, [\alpha]))$
10:    for $i = 1 : n$ do
11:        for $j = 1 : m$ do
12:            $\pi_j \leftarrow RL(\theta_i)$
13:            $\beta_{ij} \leftarrow p[j] \cdot \prod_i \pi_j(a_i | s_i)P(s_{i+1} | s_i, a_i)$ where $s_i, a_i \in \tau^i$
14:            $\beta_j \leftarrow \frac{\beta_{ij}}{\sum_{j} \beta_{ij}}$ ▶ normalize $\beta_j$
15:            $nc = nc - \beta_i$ ▶ element wise subtraction
16:        $\beta_i \leftarrow \text{bootstrap}(\beta_i)$ ▶ weighted re-sampling from the distribution $\beta_i$
17:        $nc = nc + \beta_i$ ▶ resest $i^{th}$ demonstration according to new distribution
18:        $nc$ = $\text{sparsify}(\cdot)$ ▶ remove zero entries
19:    for $j = 1 : m$ do
20:        $D_j \leftarrow \text{visitationFromDemonstration}(\mathcal{D}, \beta_i)$
21:        $D_{nj} \leftarrow \text{visitationFromPolicy}(\pi_j, ss, \beta_j)$
22:        $\phi_D \leftarrow \sum_{s,a} D_j(s,a)\phi(s,a)
23:        $\phi_n \leftarrow \sum_{s,a} D_{nj}(s,a)\phi(s,a)$
24:        $\theta_j \leftarrow \theta_j - \alpha (\phi_D - \phi_n)$
25:        $\Psi(\theta_j) \leftarrow \sum_{2}$
26:        if $\| \phi_D - \phi_n \|_\infty < \text{threshold}$ then ▶ if the gradient $\to 0$
27:            return $\theta$
Always some non-zero probability of creating a new cluster
Algorithm - BCIRL

Always some non-zero probability of creating a new cluster

For every demonstration-cluster combination, compute $\beta_{ij}$
Algorithm - BCIRL

\[ \nabla_{w_j} \mathcal{L}(\Theta, \Psi) = \sum_i \beta_{ij} \left( \mu(\tau^i) - \mu(\pi_j) \right) \]

Always some non-zero probability of creating a new cluster

Weighted resampling to avoid having unlikely clusters (just like particle filters)

**Algorithm 4 non-parametric BCIRL**

1. \( n \leftarrow \text{Number of demonstrations in } \mathcal{D} \)
2. \( \text{nc} \leftarrow \square \)  \( \triangleright \) Number of demonstrations in each cluster
3. \( P \leftarrow \text{The transition probability of the MDP} \)
4. \( \beta_i \leftarrow 0 \ \forall i = 1 : n \)
5. \( \text{ss} \leftarrow \text{startStates}(\mathcal{D}) \)
6. **while True do**
7. \( m \leftarrow \text{len(nc)} \)
8. \( \theta_{(m+1)} \leftarrow \text{Random Sample from prior} \)
9. \( p \leftarrow \text{normalize(merge(nc, [\theta])}) \)
10. **for** \( i = 1 : n \) **do**
11. \( \text{for } j = 1 : m \text{ do} \)
12. \( \pi_j \leftarrow RL(\theta_j) \)
13. \( \beta_{ij} \leftarrow p[j] \cdot \prod_i \pi_j(a_i|s_i)P(s_{i+1}|s_i, a_i) \text{ where } s_i, a_i \in \tau^i \)
14. \( \beta_{ij} \leftarrow \frac{\beta_{ij}}{\sum_j \beta_{ij}} \)  \( \triangleright \) normalize \( \beta_{ij} \)
15. \( \text{nc} \leftarrow \text{nc} - \beta_i \)  \( \triangleright \) element wise subtraction
16. \( \beta_i \leftarrow \text{bootStrap}(\beta_i) \)  \( \triangleright \) weighted re-sampling from the distribution \( \beta_i \)
17. \( \text{nc} \leftarrow \text{nc} + \beta_i \)  \( \triangleright \) re-set \( i^{th} \) demonstration according to new distribution
18. \( \text{nc.sparisify()} \)  \( \triangleright \) remove zero entries
19. **for** \( j = 1 : m \) **do**
20. \( D_j \leftarrow \text{visitationFromDemonstration}(\mathcal{D}, \beta_j) \)
21. \( D_{\pi_j} \leftarrow \text{visitationFromPolicy}(\pi_j, \text{ss}, \beta_j) \)
22. \( \hat{\phi}_D \leftarrow \sum_{s,a} D_j(s, a)\phi(s, a) \)
23. \( \hat{\phi}_{\pi_j} \leftarrow \sum_{s,a} D_{\pi_j}(s, a)\phi(s, a) \)
24. \( \theta_j \leftarrow \theta_j - \alpha(\hat{\phi}_D - \hat{\phi}_{\pi_j}) \)
25. \( \Psi(\theta_j) \leftarrow \sum_{s,a} \phi(s, a) \)
26. **if** \( \|\|\hat{\phi}_D - \hat{\phi}_{\pi_j}\|\|_\infty < \text{threshold} \) **then**  \( \triangleright \) if the gradient \( \rightarrow 0 \)
27. \( \text{return } \theta \)
Always some non-zero probability of creating a new cluster

Weighted resampling to avoid having unlikely clusters (just like particle filters)

Clustering happens here

Algorithm - BCIRL

\[ \nabla_{w_{ij}} \mathcal{L}(\Theta, \Psi) = \sum_{i} \beta_{ij} \left( \mu(\tau^j) - \mu(\pi_j) \right) \]

Algorithm 4: non-parametric BCIRL(\(D, \alpha, RL\))

1: \(n \leftarrow \) Number of demonstrations in \(D\)
2: \(nc \leftarrow \emptyset\)  \(\triangleright\) Number of demonstrations in each cluster
3: \(P \leftarrow \) The transition probability of the MDP
4: \(\beta_i \leftarrow 0 \quad \forall i = 1 : n\)
5: \(ss \leftarrow \text{startStates}(D)\)
6: while True do
7:     \(m \leftarrow \text{len}(nc)\)
8:     \(\theta_{(m+1)} \leftarrow \text{Random Sample from prior}\)
9:     \(p \leftarrow \text{normalize(merge(nc, [\theta]))}\)
10:    for \(i = 1 : n\) do
11:        for \(j = 1 : m\) do
12:            \(\pi_j \leftarrow RL(\theta_j)\)
13:            \(\beta_{ij} \leftarrow P[j] \cdot \prod_i \pi_j(a_i | s_i)P(s_{i+1} | s_i, a_i)\) where \(s_i, a_i \in \tau^i\)
14:        \(\beta_j \leftarrow \frac{\sum_j \beta_{ij}}{m}\) \(\triangleright\) normalize \(\beta_j\)
15:        \(nc = nc + \beta_j\) \(\triangleright\) element wise subtraction
16:        \(\beta = \text{bootstrap}(\beta)\) \(\triangleright\) weighted re-sampling from the distribution \(\beta\)
17:        \(nc = nc + \beta\) \(\triangleright\) reseat \(i^{th}\) demonstration according to new distribution
18:        \(nc = \text{sparseify}(\cdot)\) \(\triangleright\) remove zero entries
19:    for \(j = 1 : m\) do
20:        \(D_j \leftarrow \text{visitationFromDemonstration}(D, \beta_{j})\)
21:        \(D_{nj} \leftarrow \text{visitationFromPolicy}(\pi_j, ss, \beta_{j})\)
22:        \(\phi_{D} = \sum_{s,a} D_j(s,a)\phi(s,a)\)
23:        \(\phi_{nj} = \sum_{s,a} D_{nj}(s,a)\phi(s,a)\)
24:        \(\theta_j = \theta_j - \alpha(\phi_{D} - \phi_{nj})\)
25:        \(\Psi(\theta_j) = \sum_{k} \frac{1}{\beta_k}\)
26:        if \(||(\phi_{D} - \phi_{nj})||_\infty < \text{threshold}||\) then \(\triangleright\) if the gradient \(\rightarrow 0\)
27:            return \(\theta\)
Algorithm - BCIRL

\[ \nabla_{w_j} \mathcal{L}(\Theta, \Psi) = \sum_i \beta_{ij} \left( \mu(\tau^i) - \mu(\pi_j) \right) \]

Always some non-zero probability of creating a new cluster

Weighted resampling to avoid having unlikely clusters (just like particle filters)

Clustering happens here

Weighted feature expectations

Algorithm 4 non-parametric BCIRL(D, \alpha, RL)

1: \( n \leftarrow \) Number of demonstrations in \( D \)
2: \( nc \leftarrow \) \[
3: \( P \leftarrow \) The transition probability of the MDP
4: \( \beta_i \leftarrow 0 \ \forall i = 1 \ldots n \)
5: \( ss \leftarrow \) startStates(D)
6: \textbf{while} True \textbf{do}
7: \( m \leftarrow \text{len}(nc) \)
8: \( \theta_{(m+1)} \leftarrow \) Random Sample from prior
9: \( p \leftarrow \) normalize(merge(nc, [\alpha]))
10: \textbf{for} \( i = 1 \ldots n \) \textbf{do}
11: \textbf{for} \( j = 1 \ldots m \) \textbf{do}
12: \( \pi_j \leftarrow RL(\theta_j) \)
13: \( \beta_{ij} \leftarrow p[j] \prod_i \pi_j(a_i|s_i)P(s_{i+1}|s_i,a_i) \) where \( s_i, a_i \in \tau^i \)
14: \( \beta_{ij} \leftarrow \frac{\beta_{ij}}{\sum \beta_{ij}} \) \textbf{for normalize} \( \beta_{ij} \)
15: \( nc = nc + \beta_i \) \textbf{for element wise subtraction}
16: \( \beta_i \leftarrow \text{bootstrap}(\beta_i) \) \textbf{for weighted re-sampling from the distribution} \( \beta'_i \)
17: \( nc = nc + \beta_i \) \textbf{for reseat} \( i^{th} \) demonstration according to new distribution \( \beta'_i \)
18: \( nc_{.\text{sparsify}}() \) \textbf{for remove zero entries}
19: \textbf{for} \( j = 1 \ldots m \) \textbf{do}
20: \( D_j \leftarrow \text{visitationFromDemonstration}(D, \beta_i) \)
21: \( D_{\pi_j} \leftarrow \text{visitationFromPolicy}(\pi_j, ss, \beta_i) \)
22: \( \tilde{\phi}_D \leftarrow \sum_s a D_j(s,a) \phi(s,a) \)
23: \( \tilde{\phi}_{\pi_j} \leftarrow \sum_s a D_{\pi_j}(s,a) \phi(s,a) \)
24: \( \theta_j \leftarrow \theta_j - \alpha (\tilde{\phi}_D - \tilde{\phi}_{\pi_j}) \)
25: \( \Psi(\theta_j) \leftarrow \sum_{\alpha} \)
26: \textbf{if} \( \| (\tilde{\phi}_D - \tilde{\phi}_{\pi_j}) \|_\infty < \text{threshold} \) \textbf{then}
27: \( \textbf{return} \theta \) \textbf{for if the gradient } \to 0
\[ \nabla_{w_j} \mathcal{L}(\Theta, \Psi) = \sum_i \beta_{ij} \left( \mu(\tau^i) - \mu(\pi_j) \right) \]

Always some non-zero probability of creating a new cluster

Weighted resampling to avoid having unlikely clusters (just like particle filters)

Clustering happens here

Weighted feature expectations

We need not solve the complete inverse problem at every iteration!

Algorithm - BCIRL

```
Algorithm 4 non-parametric:BCIRL(D, \alpha, RL)
1: n \leftarrow \text{Number of demonstrations in } D
2: nc \leftarrow [] \quad \triangleright \text{Number of demonstrations in each cluster}
3: P \leftarrow \text{The transition probability of the MDP}
4: \beta_i \leftarrow 0 \quad \forall i = 1 : n
5: ss \leftarrow \text{startStates}(D)
6: while True do
7: \quad m \leftarrow \text{len}(nc)
8: \quad \theta_{(m+1)} \leftarrow \text{Random Sample from prior}
9: \quad p \leftarrow \text{normalize(merge(nc, [\alpha]))}
10: \quad for i = 1 : n do
11: \quad \quad \text{for } j = 1 : m do
12: \quad \quad \quad \pi_j \leftarrow RL(\theta_j)
13: \quad \quad \quad \beta_{ij} \leftarrow p[j] \cdot \prod_{i} \pi_j(a_i|s_i) P(s_{i+1}|s_i, a_i) \text{ where } s_i, a_i \in \tau^i
14: \quad \quad \quad \beta_j \leftarrow \sum_{i} \beta_{ij} \quad \triangleright \text{normalize } \beta_j
15: \quad \quad \text{nc} \leftarrow \text{nc} + \beta_j \quad \triangleright \text{element wise subtraction}
16: \quad \quad \text{beta} \leftarrow \text{bootStrap}(\beta) \quad \triangleright \text{weighted re-sampling from the distribution } \beta'
17: \quad \quad \text{nc} = \text{nc} + \beta_j \quad \triangleright \text{reset } i^{th} \text{ demonstration according to new distribution}
18: \quad \quad nc\text{-sparsify}() \quad \triangleright \text{remove zero entries}
19: \quad \text{for } j = 1 : m do
20: \quad \quad D_j \leftarrow \text{visitationFromDemonstration}(D, \beta_j)
21: \quad \quad D_{nj} \leftarrow \text{visitationFromPolicy}(\pi_j, ss, \beta_j)
22: \quad \quad \hat{\phi}_D \leftarrow \sum_{s,a} D_j(s, a) \phi(s, a)
23: \quad \quad \hat{\phi}_{nj} \leftarrow \sum_{s,a} D_{nj}(s, a) \phi(s, a)
24: \quad \quad \theta_j \leftarrow \theta_j - \alpha (\hat{\phi}_D - \hat{\phi}_{nj})
25: \quad \quad \Psi(\phi) \leftarrow \sum_{s,a} \nabla \phi(s, a)
26: \quad \quad \text{if } \| (\hat{\phi}_D - \hat{\phi}_{nj}) \|_{\infty} < \text{threshold} \quad \triangleright \text{if the gradient } \rightarrow 0
27: \quad \quad \text{return } \theta
```
Results

- On a motivating example

\[ \mathcal{D} = \left\{ \{R, R, R\} \times 50, \{L, L, L\} \times 50 \right\} \]
# Results

- On a motivating example

\[ D = \left\{ \{R, R, R\} \times 50, \{L, L, L\} \times 50 \right\} \]

<table>
<thead>
<tr>
<th></th>
<th>Policy</th>
<th>Likelihood of demonstrations (objective)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxEnt IRL</td>
<td>( P(R) = 0.53 ) ( P(L) = 0.47 )) ( P(D</td>
<td>w) = 0.13 )</td>
</tr>
<tr>
<td>Non-parametric BCIRL</td>
<td>( P_1(R) = 1 ) ( P_1(L) = 0 ) ( P_2(R) = 0 ) ( P_2(L) = 1 ) ( P(D</td>
<td>\Theta) = 1 )</td>
</tr>
</tbody>
</table>
Results

- Highway task

  Aggressive demonstrations:

  Evasive demonstrations:
Results

- Demonstrations

- Learned Behaviors
### Results

- **Demonstrations**

<table>
<thead>
<tr>
<th>Method</th>
<th>Likelihood of demonstrations (objective value) at convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxEnt IRL</td>
<td>$P(D</td>
</tr>
<tr>
<td>Non-parametric BCIRL</td>
<td>$P(D</td>
</tr>
</tbody>
</table>

$0.9^{40} = 0.015$
Results

- Demonstrations

<table>
<thead>
<tr>
<th>Method</th>
<th>Likelihood of demonstrations (objective value) at convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxEnt IRL</td>
<td>$P(D</td>
</tr>
<tr>
<td>Non-parametric BCIRL</td>
<td>$P(D</td>
</tr>
</tbody>
</table>

- Ref: $0.9^{40} = 0.015$
Results

- Gazebo simulator
Results

- Gazebo simulator
  - Aggressive behavior using potential field controller
Results

- Gazebo simulator
  - Aggressive behavior using potential field controller
Results

- Discretize the state space based on size of the car
- Gazebo simulator results

<table>
<thead>
<tr>
<th>Method</th>
<th>Likelihood of demonstrations (objective value) at convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxEnt IRL</td>
<td>$P(\mathcal{D}</td>
</tr>
<tr>
<td>Non-parametric BCIRL</td>
<td>$P(\mathcal{D}</td>
</tr>
</tbody>
</table>
## Results

- **Gazebo simulator results**

<table>
<thead>
<tr>
<th></th>
<th>Likelihood of demonstrations (objective value) at convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxEnt IRL</td>
<td>$P(D</td>
</tr>
<tr>
<td>Non-parametric BCIRL</td>
<td>$P(D</td>
</tr>
</tbody>
</table>

- **Clustered into - [21, 19, 5, 4, 1]**
Results

- **Gazebo simulator results**

<table>
<thead>
<tr>
<th></th>
<th>Likelihood of demonstrations (objective value) at convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxEnt IRL</td>
<td>$P(\mathcal{D}</td>
</tr>
<tr>
<td>Non-parametric BCIRL</td>
<td>$P(\mathcal{D}</td>
</tr>
</tbody>
</table>

- Clustered into - [21, 19, 5, 4, 1]
  - Cluster 1: Evasive
  - Cluster 2: Aggressive
  - Clusters 3, 4, and 5: Neither

- **Able to learn the behaviors though we cannot get consistent demonstrations**
Conclusion

- **Advantages of Behavior Clustering IRL**
  - Can cluster demonstrations and learn reward function for each behavior
  - Can predict new samples with high probability
  - Can be used to separate consistent demonstrations from the rest

- **Disadvantages**
  - Feature selection is harder
    - Need features to also explain the differences in the behavior
  - Does not scale well (exists in MaxEnt also)
  - Solve multiple IRL problems for each cluster
Future work

- Addressing some of the disadvantages
  - Feature selection
    - Feature construction for IRL (Levine 2010)
    - Guided cost learning (Finn 2016)
  - Scalability (exists in MaxEnt also)
    - Guided Policy search (Levine 2013)
    - Path integral and Metropolis Hasting sampling (Kappen 2009)
Part II
Approximate Optimal Control with Temporal Logic Tasks
Outline

- Background
  - LTL specifications
  - Reward shaping
  - Policy Gradients
  - Actor critic

- Method
  - Relation between Reward shaping and Actor critic
  - Heuristic value initialization

- Results

- Conclusion
Motivation

- Motivating example
  - Robot Soccer

Robot Soccer. Source: IEEE Spectrum, Internet
Motivation

- Simpler task
  - No opponents or teammates
  - Robot Soccer
    - There is a sequence of requirements temporally constrained
  - For example,
    - Get the ball - T1
    - Go near the goal - T2
    - Shoot - T3
  - LTL specification

\[ \Diamond (T_1 \land \Diamond (T_2 \land \Diamond T_3)) \]

\[ \Diamond \text{ – eventually} \]
\[ \land \text{ – and} \]
Motivation

- Simpler task
  - No opponents or teammates
- Define the reward function
  - +1 if Goal is scored
Motivation - Why just RL fails

- Simpler task
  - No opponents or teammates
- Define the reward function
  - +1 if Goal is scored
    - Very hard to explore
Motivation

- How to use LTL to accelerate
- LTL specification
  - Either True when satisfied
  - Or False otherwise
  - There is no signal towards partial completion
- Exploit structure in actor critic to motivate the agent towards completion
Preliminaries
Reward Shaping

- Simpler task
  - No opponents or teammates
- Define the reward function
  - +1 if Goal is scored

- We need to satisfy temporally related requirements
  - Get the ball: $R = 0.01$ (shaping reward)
  - Score a Goal: $R = +1$ (true reward)
**Reward Shaping**

- We need to satisfy temporally related requirements
  - Example,
    - Get the ball: $R = 0.01$ (shaping reward)
    - Score a Goal: $R = +1$ (true reward)
Reward Shaping

- We need to satisfy temporally related requirements
  - Example,
    - Get the ball: $R = 0.01$ (shaping reward)
    - Score a Goal: $R = +1$ (true reward)
  - Result (Andrew Ng 1999)
    - The agent keeps vibrating near the ball
Reward Shaping

- Before shaping
  - Optimal policy was to score a goal
- After shaping
  - Optimal policy is to vibrate near the ball
- Optimal policy Invariance (Andrew Ng 1999)

\[
\hat{r}(s, a, s') = r(s, a, s') + \xi(s') - \xi(s)
\]

\(\hat{r}\)– shaping term
Reward Shaping

- Before shaping
  - Optimal policy was to score a goal
- After shaping
  - Optimal policy is to vibrate near the ball
- Optimal policy Invariance (Andrew Ng 1999)

\[
\hat{r}(s, a, s') = r(s, a, s') + \xi(s') - \xi(s)
\]

\[
\hat{R}(\tau_{closed}) = \hat{r}_0 + \hat{r}_1 + \cdots + \hat{r}_n
\]

\[
\hat{R}(\tau_{closed}) = (\xi(s_1) - \xi(s_0)) + (\xi(s_2) - \xi(s_1)) + \cdots + (\xi(s_n) - \xi(s_0))
\]
Reward Shaping and Policy Invariance

- **Before shaping**
  - Optimal policy was to score a goal
- **After shaping**
  - Optimal policy is to vibrate near the ball
- **Optimal policy Invariance (Andrew Ng 1999)**

\[
\hat{r}(s, a, s') = r(s, a, s') + \underbrace{\xi(s') - \xi(s)}_{\hat{r} - \text{shaping term}}
\]

\[
\hat{R}(\tau_{\text{closed}}) = \underbrace{(\xi(s_1) - \xi(s_0))}_{\hat{r}_0} + \underbrace{(\xi(s_2) - \xi(s_1))}_{\hat{r}_1} + \cdots + \underbrace{(\xi(s_n) - \xi(s_0))}_{\hat{r}_n}
\]

\[
= (\xi(s_1) - \xi(s_0)) + (\xi(s_2) - \xi(s_1)) + \cdots + (\xi(s_n) - \xi(s_0))
\]

\[
= 0
\]
Before shaping
- Optimal policy was to score a goal

After shaping
- Optimal policy is to vibrate near the ball

Optimal policy Invariance (Andrew Ng 1999)
- More generally

\[ \hat{r}(s, a, s') = r(s, a, s') + \gamma \xi(s') - \xi(s) \]

\(\hat{r}\)—shaping term
Preliminaries - Policy Gradients

Objective of RL

\[ \max \pi \mathbb{E}_{\tau \sim \pi} [R(\tau)] \]
Preliminaries - Policy Gradients

Objective of RL

$$\max_{\pi} E_{\tau \sim \pi}[R(\tau)]$$

By parametrizing the policy

$$\max_{\theta} E_{\tau \sim \pi_{\theta}}[R(\tau)]$$

$$R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots$$
Objective of RL

\[
\max_{\pi} E_{\pi \sim \pi}[R(\tau)]
\]

By parametrizing the policy

\[
\max_{\theta} E_{\pi \sim \pi_\theta}[R(\tau)]
\]

Utility of the parameter

\[
U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)
\]

Objective: \( \max_{\theta} U(\theta) \)
Preliminaries - Policy Gradients

- Gradient of the utility from samples

\[ R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \]

\[ U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau) \]
Preliminaries - Policy Gradients

- Gradient of the utility from samples

\[ R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \]
\[ U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau) \]
Preliminaries - Policy Gradients

- Gradient of the utility from samples

\[ R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \]

\[ U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau) \]

- Policy gradient

\[ \nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \left[ \nabla_\theta \ln \pi(a_t^{(i)}|s_t^{(i)}; \theta) \left( \sum_{k=t}^{T-1} \gamma^{t-k} r(s_k^{(i)}, a_k^{(i)}) \right) \right] \]
Background - Actor Critic

- Policy gradients

\[
\nabla_{\theta} U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \nabla_{\theta} \ln \pi(a_t^{(i)} | s_t^{(i)}; \theta) \left( \sum_{k=t}^{T-1} \gamma^{t-k} r(s_k^{(i)}, a_k^{(i)}) \right)
\]

\[
R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots
\]

\[
U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)
\]

\[
V(s) = E[R(\tau)|s_t = s]
\]

\[
\tau^{(i)} = \{(s_0^{(i)}, a_0^{(i)}, r_0^{(i)}), (s_1^{(i)}, a_1^{(i)}, r_1^{(i)}) \cdots \}
\]
Background - Actor Critic

- **Policy gradients**

\[ \nabla_{\theta} U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \left[ \nabla_{\theta} \ln \pi(a_{t}^{(i)} | s_{t}^{(i)} ; \theta) \left( \sum_{k=t}^{T-1} \gamma^{t-k} r(s_{k}^{(i)}, a_{k}^{(i)}) \right) \right] \]

- **Reward shaping**

  replace \( r(s, a, s') \) with \( r(s, a) + \gamma \xi(s') - \xi(s) \)

\[
R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \\
U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau) \\
V(s) = E[R(\tau) | s_{t} = s] \\
\tau^{(i)} = \{ (s_{0}^{(i)}, a_{0}^{(i)}, r_{0}^{(i)}), (s_{1}^{(i)}, a_{1}^{(i)}, r_{1}^{(i)}) \cdots \} \]
Background - Actor Critic

- **Policy gradients**

\[ \nabla_{\theta} U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \left[ \nabla_{\theta} \ln \pi(a_t^{(i)}|s_t^{(i)}; \theta) \left( \sum_{k=t}^{T-1} \gamma^{t-k} r(s_k^{(i)}, a_k^{(i)}) \right) \right] \]

- **Reward shaping**

  replace \( r(s, a, s') \) with \( r(s, a) + \gamma \xi(s') - \xi(s) \)

- **Actor Critic**

  \[ \xi = V \]
Background - Actor Critic

- **Policy gradients**

\[
\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \left[ \nabla_\theta \ln \pi(a_t^{(i)} | s_t^{(i)}; \theta) \left( \sum_{k=t}^{T-1} \gamma^{t-k} r(s_k^{(i)}, a_k^{(i)}) \right) \right]
\]

- **Reward shaping**

replace \( r(s, a, s') \) with \( r(s, a) + \gamma \xi(s') - \xi(s) \)

- **Actor Critic**

\[\xi = V\]

\[
\hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \left[ \nabla_\theta \ln \pi(a_t^{(i)} | s_t^{(i)}; \theta) \left( \sum_{k=t}^{T-1} \gamma^{t-k} r(s_k^{(i)}, a_k^{(i)}) - V(s_t^{(i)}) \right) \right]
\]

Increases the probability of \( a_t^{(i)} \)

Amount of increase (can be negative)
Background - Actor Critic

- **Policy gradients**

\[
\nabla_{\theta} U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \left[ \nabla_{\theta} \ln \pi(a_t^{(i)}|s_t^{(i)}; \theta) \left( \sum_{k=t}^{T-1} \gamma^{t-k}r(s_k^{(i)}, a_k^{(i)}) \right) \right]
\]

- **Reward shaping**

replace \( r(s, a, s') \) with \( r(s, a) + \gamma \xi(s') - \xi(s) \)

- **Actor Critic**

\[
\xi = V
\]

\[
\hat{g} = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T-1} \left[ \nabla_{\theta} \ln \pi(a_t^{(i)}|s_t^{(i)}; \theta) \left( r(s_t^{(i)}, a_t^{(i)}) + \gamma V(s_{t+1}^{(i)}) - V(s_t^{(i)}) \right) \right]
\]

Amount of increase (can be negative)

\[
R(\tau) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots
\]

\[
U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)
\]

\[
V(s) = E[R(\tau)|s_t = s]
\]

\[
\tau^{(i)} = \{ (s_0^{(i)}, a_0^{(i)}, r_0^{(i)}), (s_1^{(i)}, a_1^{(i)}, r_1^{(i)}), \ldots \}
\]
Background - Actor Critic

- Actor Critic
  - Actor (policy) update
    - Use $\hat{g}$ the empirical estimate of the gradient
  - Critic (value) update
    - Use any supervised learning to learn the targets $V(s) = E[R(\tau)|s_t = s]$
Background - Actor Critic

- Actor Critic
  - Actor (policy) update
    - Use $\hat{g}$ the empirical estimate of the gradient
  - Critic (value) update
    - Use any supervised learning to learn the targets $V(s) = E[R(\tau) | s_t = s]$
  - Critics are shaping functions
Method - Accelerating Actor Critic using LTL

- Given a specification
  \[ \varphi = (\Diamond(R_1 \land R_3) \lor \Diamond(R_2 \land R_3)) \land \Box \neg O \]
- +10 for satisfying the specification

- Agent
- R1
- R2
- R3
- O

<table>
<thead>
<tr>
<th>R1</th>
<th>Region 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>Region 2</td>
</tr>
<tr>
<td>R3</td>
<td>Region 3</td>
</tr>
<tr>
<td>O</td>
<td>Obstacles</td>
</tr>
</tbody>
</table>
Method - Accelerating Actor Critic using LTL

- Given a specification
  \[ \varphi = (\Diamond (R_1 \land R_3) \lor \Diamond (R_2 \land R_3)) \land \Box \neg O \]

- Break down into several reach avoid task for critic initialization
  - Task1:
    \[ \varphi_1 = \Diamond (R_1 \lor R_2) \land \Box \neg O \]
  - Task2:
    \[ \varphi_2 = \Diamond R_3 \land \Box \neg O \]

<table>
<thead>
<tr>
<th></th>
<th>Region 1</th>
<th>Region 2</th>
<th>Region 3</th>
<th>Obstacles</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Diamond)</td>
<td>eventually</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\land)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\lor)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\Box)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\neg)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Break down into several reach avoid task
  • Task1:
    \[ \varphi_1 = \Diamond (R_1 \lor R_2) \land \Box \neg O \]
  • Task2:
    \[ \varphi_2 = \Diamond R_3 \land \Box \neg O \]
• Automata of the original specification

\[ \varphi = (\Diamond (R_1 \land R_3) \lor \Diamond (R_2 \land R_3)) \land \Box \neg O \]

\[ \Diamond \] – eventually
\[ \land \] – and
\[ \lor \] – or
\[ \Box \] – always
\[ \neg \] – not

R_1 – Region 1
R_2 – Region 2
R_3 – Region 3
O – Obstacles
E – Everything else
Method - Accelerating Actor Critic using LTL

- Break down into several reach avoid task
  - Task1:
    \[ \varphi_1 = \Diamond (R_1 \lor R_2) \land \Box \neg O \]
  - Task2:
    \[ \varphi_2 = \Diamond R_3 \land \Box \neg O \]
- Heuristic value initialization

\[ \hat{V}(s, q) = \max_i (\gamma \|s - g_i(q)\|_1) \]

\( \gamma \in [0, 1) \)
**Method - Accelerating Actor Critic using LTL**

- Heuristic value initialization for Task 2
  \[
  \hat{V}(s, q) = \max_i (\gamma^{s-g_i(q)}_1)
  \]

- Reward: +10 if \( \varphi \) is satisfied. -5 for running into obstacles.

\[
\varphi_1 = \Diamond(R_1 \lor R_2) \land \Box \neg O
\]

\[
\varphi_2 = \Diamond R_3 \land \Box \neg O
\]
Results

- Learned values
- Reward: +10 if $\varphi$ is satisfied. -5 for running into obstacles.

$\varphi_1 = \Diamond (R_1 \lor R_2) \land \Box \neg O$

$\varphi_2 = \Diamond R_3 \land \Box \neg O$
Results

- Actor critic with and without heuristic initialization
- Reward: +10 if $\varphi$ is satisfied. -5 for running into obstacles.
Conclusion and Discussions

- **Summary**
  - IRL with automated behavior clustering
    - Improve feature selection and scalability
  - Accelerating actor-critic with temporal logic constraints.
    - Automate the decomposition procedure for LTL specifications for scalable systems
- **Possible directions**
  - Use LTL specifications to accelerate BCIRL
- **Applications to general domains:**
  - Big data
  - Urban planning