Imitation Learning

Goal:

- Learn how to make decisions by trying to imitate another agent.
Imitation Learning

Goal:

- Learn how to make decisions by trying to imitate another agent.

Conventional Imitation Learning:

- Observations of other agent (demonstrations) consist of state-action pairs.\(^1\)

---

\(^1\) Niekum et al., “Learning and generalization of complex tasks from unstructured demonstrations”.
Imitation Learning

Goal:
- Learn how to make decisions by trying to imitate another agent.

Conventional Imitation Learning:
- Observations of other agent (demonstrations) consist of state-action pairs.¹
Imitation Learning

Goal:
- Learn how to make decisions by trying to imitate another agent.

Conventional Imitation Learning:
- Observations of other agent (demonstrations) consist of state-action pairs.¹

Challenge:
- Precludes using a large amount of demonstration data where action sequences are not given (e.g. YouTube videos).

¹ Niekum et al., “Learning and generalization of complex tasks from unstructured demonstrations”.
Imitation Learning

Algorithms:
Imitation Learning

Algorithms:

- Behavioral Cloning:
Imitation Learning

Algorithms:

- Behavioral Cloning:
  - End to End Learning for Self-Driving Cars.²

² Zhang and Cho, “Query-Efficient Imitation Learning for End-to-End Simulated Driving.”
Imitation Learning

Algorithms:

- Behavioral Cloning:
  - End to End Learning for Self-Driving Cars.²

- Inverse Reinforcement Learning:

² Zhang and Cho, “Query-Efficient Imitation Learning for End-to-End Simulated Driving.”
Imitation Learning

Algorithms:

- Behavioral Cloning:
  - End to End Learning for Self-Driving Cars.\(^2\)

- Inverse Reinforcement Learning:
  - Generative Adversarial Imitation Learning.\(^3\)
  - Guided Cost Learning.\(^4\)

---

\(^2\) Zhang and Cho, “Query-Efficient Imitation Learning for End-to-End Simulated Driving.”

\(^3\) Ho and Ermon, “Generative adversarial imitation learning”.

\(^4\) Finn, Levine, and Abbeel, “Guided cost learning: Deep inverse optimal control via policy optimization.”
Imitation from Observation

Goal:

- Learn how to perform a task given state-only demonstrations.
Imitation from Observation

Goal:

- Learn how to perform a task given state-only demonstrations.
Imitation from Observation

Goal:
- Learn how to perform a task given state-only demonstrations.

Formulation:
- Given:
  - $D_{demo} = (s_0, s_1, ...)$
- Learn:
  - $\pi : S \rightarrow A$
Imitation from Observation

Previous work:
Imitation from Observation

Previous work:

- Time Contrastive Networks (TCN).\(^5\)
- Imitation from observation: Learning to imitate behaviors from raw video via context translation.\(^6\)
- Learning invariant feature spaces to transfer skills with reinforcement learning.\(^7\)

---


\(^6\) Liu et al., “Imitation from observation: Learning to imitate behaviors from raw video via context translation”.

\(^7\) Gupta et al., “Learning invariant feature spaces to transfer skills with reinforcement learning”.
Imitation from Observation

Previous work:

- Time Contrastive Networks (TCN).\(^5\)
- Imitation from observation: Learning to imitate behaviors from raw video via context translation.\(^6\)
- Learning invariant feature spaces to transfer skills with reinforcement learning.\(^7\)

Concentrate on perception; require time-aligned demonstrations.

---


\(^6\) Liu et al., “Imitation from observation: Learning to imitate behaviors from raw video via context translation”.

\(^7\) Gupta et al., “Learning invariant feature spaces to transfer skills with reinforcement learning”.
Efficient Robot Skill Learning

- Motivation: RoboCup
- Sim2Real: Grounded Simulation Learning
- Imitation Learning from Observation:
  - Model-based approach: BCO
  - Model-free approach: GAIfO
Model-based Approach

- Imitation Learning: \[ D_{\text{demo}} = \{(s_0, a_0), (s_1, a_1), \ldots\} \]
Model-based Approach

- Imitation Learning: \( D_{demo} = \{(s_0, a_0), (s_1, a_1), \ldots\} \)
- Imitation from Observation: \( D_{demo} = \{(s_0, ?), (s_1, ?), \ldots\} \)
Model-based Approach

- Imitation Learning: \( D_{demo} = \{(s_0, a_0), (s_1, a_1), \ldots\} \)
- Imitation from Observation: \( D_{demo} = \{(s_0, ?), (s_1, ?), \ldots\} \)

Model-based Approach:

1. Learn an inverse dynamics model
2. Infer actions
3. Perform a conventional IL method
Behavioral Cloning from Observation (BCO)

Algorithm:

START

1. Initialize policy $\pi_\phi$
2. Run policy $\pi_\phi$
3. Collect data $T_\pi, A_\pi$
4. Update model $M_\theta$
5. State-only demonstrations

$\pi_\phi^0 \xrightarrow{\pi_\phi} \{(s_t^a, s_{t+1}^a)\} \xrightarrow{a_t} T_\pi, A_\pi \xrightarrow{\tilde{A}_{demo}} S_{demo} \xrightarrow{M_\theta} D_{demo}$

Torabi, Warnell, and Stone, IJCAI 2018
Behavioral Cloning from Observation (BCO)

Experimental Results:
- **Domain:**
  - Mujoco domain "Ant" with 111 dimensional state space and 8 dimensional action space.
Behavioral Cloning from Observation (BCO)

Experimental Results:

![Graph showing performance versus number of demonstrated trajectories for different methods: Random, Expert, GAIL, BCO(0), FEM, BC.]
Behavioral Cloning from Observation (BCO)

Experimental Results:

![Graph showing performance against number of demonstrated trajectories. The graph includes lines for Random, Expert, GAIL, BCO(0), FEM, and BC. The x-axis represents the number of demonstrated trajectories (5, 10, 15, 20, 25), and the y-axis represents performance ranging from -1.5 to 1.0. The expert trajectory is marked with a solid black line, while other methods are represented with different colored lines.]
Behavioral Cloning from Observation (BCO)

Experimental Results:

![Graph showing experimental results for Ant with different algorithms. The x-axis represents the number of demonstrated trajectories, while the y-axis shows performance. The algorithms compared are Random, GAIL, Expert, BCO(0), FEM, and BC. The bars indicate performance with error bars for each algorithm at different trajectory counts.]
Behavioral Cloning from Observation (BCO($\alpha$))

Issue:
- Inverse dynamics model is learned using a random policy.
Behavioral Cloning from Observation (BCO(α))

Issue:
  - Inverse dynamics model is learned using a random policy.

Solution: BCO(α)
Behavioral Cloning from Observation (BCO(\(\alpha\)))

Issue:
- Inverse dynamics model is learned using a random policy.

Solution: BCO(\(\alpha\))
- Update the model with the learned policy.
Behavioral Cloning from Observation (BCO(\(\alpha\)))

Issue:
- Inverse dynamics model is learned using a random policy.

Solution: BCO(\(\alpha\))
- Update the model with the learned policy.
- Parameter \(\alpha\) controls tradeoff between performance and environment interactions
Behavioral Cloning from Observation (BCO(\(\alpha\)))

Issue:
- Inverse dynamics model is learned using a random policy.

Solution: BCO(\(\alpha\))
- Update the model with the learned policy.
- Parameter \(\alpha\) controls tradeoff between performance and environment interactions
  - \(\alpha = 0\): no post-demonstration interaction.
Behavioral Cloning from Observation (BCO(α))

Issue:
- Inverse dynamics model is learned using a random policy.

Solution: BCO(α)
- Update the model with the learned policy.
- Parameter $\alpha$ controls tradeoff between performance and environment interactions
  - $\alpha = 0$: no post-demonstration interaction.
  - Increasing $\alpha$: increasing the number of interactions allowed at each iteration.
Behavioral Cloning from Observation (BCO(\(\alpha\)))

Algorithm:

1. **START**
   - Initialize policy \(\pi_\phi\)
   - State-only demonstrations

2. **Run policy \(\pi_\phi\)**
   - \(\pi_\phi^0\)
   - \(\{(s_t^\alpha, s_{t+1}^\alpha)\}\)
   - \(\{a_t\}\)

3. **Collect data**
   - \(T_\pi^\alpha, A_\pi\)

4. **Update model \(M_\theta\)**
   - \(M_\theta\)
   - \(\tilde{A}_{demo}\)

5. **Infer actions**
   - \(S_{demo}\)
   - \(D_{demo}\)
Behavioral Cloning from Observation (BCO(α))

Algorithm:

START

Initialize policy $\pi_{\phi}^{i=0}$

Run policy $\pi_{\phi}^{i}$

Append to $T_{\pi}^{a}, A_{\pi}$

Update model $M_{\theta}^{i}$

State-only demonstrations

Behavioral Cloning from Observation (BCO)

$\mathcal{D}_{demo}$

$(s_t^a, s_{t+1}^a)$

$\{a_t\}$
Behavioral Cloning from Observation (BCO(α))

Interaction time:

- **Pre-demonstration**
  - BCO(0)
  - BCO(α)
  - GAIL & FEM

- **Post-demonstration**
  - Environment Interactions
  - Inverse Model Update
  - Policy Learning Update

| I_{pre} | I_{pre} | I_{pre} | α_{T_{pre}} | α_{T_{pre}} | α_{T_{pre}} | ... | I_{IRL} |
Behavioral Cloning from Observation (BCO($\alpha$))

Effect of varying $\alpha$ on BCO($\alpha$):

![Graph showing the effect of varying $\alpha$ on BCO($\alpha$). The x-axis represents the number of demonstrated trajectories, and the y-axis represents performance. Different lines indicate different values of $\alpha$ and the performance level.](image-url)
Behavioral Cloning from Observation (BCO($\alpha$))

Effect of varying $\alpha$ on BCO($\alpha$):
Behavioral Cloning from Observation (BCO(\(\alpha\)))

Effect of varying \(\alpha\) on BCO(\(\alpha\)):

![Graph showing the effect of varying \(\alpha\) on BCO(\(\alpha\)).](image)
Efficient Robot Skill Learning

- Motivation: RoboCup
- Sim2Real: Grounded Simulation Learning
- Imitation Learning from Observation:
  - Model-based approach: BCO
  - Model-free approach: GAIfO
Gen. Adversarial Imitation from Observation (GAIfO)

Motivation:

(a) Random Policy

(b) Demonstration

Figure: State transition distribution in Hopper domain.
Gen. Adversarial Imitation from Observation (GAIfO)

Algorithm:

Demonstrator

\[ \nu \uparrow 1 \]

\[ D \]

\[ s \rightarrow \cdots \rightarrow s' \]

Imitator

\[ \nu \downarrow 0 \]

\[ D \]

\[ s \rightarrow \cdots \rightarrow s' \]

Environment

\[ s \rightarrow \cdots \rightarrow a \rightarrow \pi \rightarrow s \]