DAgger: Taming Covariate Shift with No Regret

Sanjiban Choudhury
Behavior Cloning crashes into a wall

Train ≠ Test
Train Test Mismatch

Training / Validation Loss

\[
\sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^*}} [\ell(s_t, \pi(s_t))] = O(\epsilon T)
\]

Test Loss

\[
\sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^\pi} [\ell(s_t, \pi(s_t))] = O(\epsilon T^2)
\]
Can we mathematically quantify how much worse BC is compared to the demonstrator?
First, let’s define **performance** of a policy

\[
J(\pi) = \mathbb{E}_{a_t \sim \pi(s_t)} \left[ \sum_{t=0}^{T-1} c(s_t, a_t) \right]
\]

(Performance)
Second, let’s define performance difference

\[ J(\pi) - J(\pi^*) \]

(Performance of my learner) (Performance of my demonstrator)

We want to *minimize* the performance difference
Behavior cloning hits the worst case!

There exists an MDP where BC has a performance difference of $O(\epsilon T^2)$

We are going to such a MDP right now, and you will see more in A1!
A Tree MDP
The demonstrator always takes a left
Assume the following cost function

\[ c(s, a) = \mathbb{I}(a \neq \pi^*(s)) \]

(0 if you agree with expert, 1 otherwise)
Assume the following cost function

\[
c(s, a) = \mathbb{I}(a \neq \pi^*(s))
\]

(0 if you agree with expert, 1 otherwise)

Note that you never see what the expert prefers in other states
Show that BC has a performance difference of $O(\epsilon T^2)$

$c(s, a) = \mathbb{1}(a \neq \pi^*(s))$

(0 if you agree with expert, 1 otherwise)
Proof
So, it seems BC is totally doomed ...
But, BC works surprisingly often!!

<table>
<thead>
<tr>
<th>Environment</th>
<th>Expert</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CartPole</td>
<td>500 ± 0</td>
<td>500 ± 0</td>
</tr>
<tr>
<td>Acrobat</td>
<td>−71.7 ± 11.5</td>
<td>−78.4 ± 14.2</td>
</tr>
<tr>
<td>MountainCar</td>
<td>−99.6 ± 10.9</td>
<td>−107.8 ± 16.4</td>
</tr>
<tr>
<td>Hopper</td>
<td>3554 ± 216</td>
<td>3258 ± 396</td>
</tr>
<tr>
<td>Walker2d</td>
<td>5496 ± 89</td>
<td>5349 ± 634</td>
</tr>
<tr>
<td>HalfCheetah</td>
<td>4487 ± 164</td>
<td>4605 ± 143</td>
</tr>
<tr>
<td>Ant</td>
<td>4186 ± 1081</td>
<td>3353 ± 1801</td>
</tr>
</tbody>
</table>

[SCV+ arXiv ’21]

[Rajeswaran et al. ’17]

[Florence et al. ’21]

[OfflineRL] [BC]
But, BC works surprisingly often!!

Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation

Tianhao Zhang\textsuperscript{1,2}, Zoe McCarthy\textsuperscript{4}, Owen Jow\textsuperscript{1}, Dennis Lee\textsuperscript{1}, Xi Chen\textsuperscript{12}, Ken Goldberg\textsuperscript{1}, Pieter Abbeel\textsuperscript{1-4}

![Image: Virtual Reality teleoperation in action](Fig. 1: Virtual Reality teleoperation in action)

---

On Bringing Robots Home

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nur Muhammad (Mohd) Shaffullah</td>
<td>NYU</td>
<td></td>
</tr>
<tr>
<td>Anas Ral</td>
<td>NYU</td>
<td></td>
</tr>
<tr>
<td>Haritha Eshkuru</td>
<td>NYU</td>
<td></td>
</tr>
<tr>
<td>Yiyan Liu</td>
<td>NYU</td>
<td></td>
</tr>
<tr>
<td>Ishan Misra</td>
<td>Meta</td>
<td></td>
</tr>
<tr>
<td>Soumil Chintala</td>
<td>Meta</td>
<td></td>
</tr>
<tr>
<td>Lerrel Pinto</td>
<td>NYU</td>
<td></td>
</tr>
</tbody>
</table>

Mission: teach Debb E to open drawers New Jersey

Collect 24 demos 5 minutes

Fine-tune model 15 minutes

Deploy!
Why does BC work in these cases?

\[ O(\epsilon T^2) \]

Drive \( \epsilon \) to 0!

When can we actually do this?
The Realizable Setting

With infinite data and a realizable expert, can drive $\epsilon \to 0$
Realizable settings are easy ...
Why is the expert realizable here? (Easy)

<table>
<thead>
<tr>
<th>Environment</th>
<th>Expert</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CartPole</td>
<td>500 ± 0</td>
<td>500 ± 0</td>
</tr>
<tr>
<td>Acrobat</td>
<td>-71.7 ± 11.5</td>
<td>-78.4 ± 14.2</td>
</tr>
<tr>
<td>MountainCar</td>
<td>-99.6 ± 10.9</td>
<td>-107.8 ± 16.4</td>
</tr>
<tr>
<td>Hopper</td>
<td>3554 ± 216</td>
<td>3258 ± 396</td>
</tr>
<tr>
<td>Walker2d</td>
<td>5496 ± 89</td>
<td>5349 ± 634</td>
</tr>
<tr>
<td>HalfCheetah</td>
<td>4487 ± 164</td>
<td>4605 ± 143</td>
</tr>
<tr>
<td>Ant</td>
<td>4186 ± 1081</td>
<td>3353 ± 1801</td>
</tr>
</tbody>
</table>

[SCV+ arXiv '21]

[Rajeswaran et al. '17]

[Florence et al. '21]
Why is the expert realizable here? (Easy)
What is the hard case where $\epsilon > 0$?

Non-realizable Expert!
Poll
Give examples of a non-realizable expert

Hint: There are at least 3 categories!

When poll is active respond at PollEv.com/sc2582
Idea for a New Algorithm!

What if we just queried the expert for the best action on states the learner visits?
Interactive Behavior Cloning

Initialize with a random policy $\pi_1$  

For $i = 1, \ldots, N$

Execute policy $\pi_i$ in the real world and collect data

$$\mathcal{D}_i = \{ s_0, a_0, s_1, a_1, \ldots \}$$

# Also called a rollout

Query the expert for the optimal action on learner states

$$\mathcal{D}_i = \{ s_0, \pi^*(s_0), s_1, \pi^*(s_1), \ldots \}$$

Train a new learner on this dataset

$$\pi_{i+1} \leftarrow \text{Train}(\mathcal{D}_i)$$
Does Interactive BC solve our Tree MDP?

Let's assume depth 2

Expert always take left

Assume every iteration $\pi_{i+1} \leftarrow \text{Train}(\mathcal{D}_i)$
we can drive down loss to $\epsilon$

Let's walk through how interactive BC does!
Interactive BC is also $O(\epsilon T^2)$!

\[ \pi_{i+1} \leftarrow \text{Train}(\mathcal{D}_i) \]

$\pi_{i+1}$ can have a totally different distribution than $\mathcal{D}_i$ generated by $\pi_i$
Instead of throwing out the old dataset, what if we aggregated data over iterations?
DAgger (Dataset Aggregation)

Initialize with a random policy $\pi_1$  # Can be BC
Initialize empty data buffer $\mathcal{D} \leftarrow \{\}$
For $i = 1, \ldots, N$

Execute policy $\pi_i$ in the real world and collect data
$$\mathcal{D}_i = \{s_0, a_0, s_1, a_1, \ldots\}$$  # Also called a rollout

Query the expert for the optimal action on learner states
$$\mathcal{D}_i = \{s_0, \pi^*(s_0), s_1, \pi^*(s_1), \ldots\}$$

Aggregate data $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$

Train a new learner on this dataset $\pi_{i+1} \leftarrow \text{Train}(\mathcal{D})$

Select the best policy in $\pi_{1:N+1}$
Why does DAgger work?

Theory of Online Learning explains why

(Next Lecture!)