A Reduction of Imitation Learning and Structure Prediction to No-Regret Online Learning
Stephane Ross, Geoffrey J. Gordon and J. Andrew Bagnell [1]

Dennis Aumiller
Heidelberg University
Imitation Learning

October 12, 2018
# Table of Contents

1. Motivation
2. Algorithm Descriptions
3. Analysis
4. Experimental Results
5. Limitations of the DAgger Algorithm
6. Conclusion
Outline

1 Motivation

2 Algorithm Descriptions

3 Analysis

4 Experimental Results

5 Limitations of the DAgger Algorithm

6 Conclusion
“Standard Arguments” for Imitation Learning:

- Generally helpful in sequential prediction problems
- Learn a robust policy that can recover from failure (compare Supervised Learning)
- *Efficiently* learn such a policy (compare Reinforcement Learning)

Further, we have seen shortcomings of previous algorithms:

- Convergence might not be (or only weakly) guaranteed
  - SEARN: Grows quadratically in the number of errors
- Resulting policy might be a stochastic mixture of several policies, or non-stationary
Application Examples

- Autonomous navigation
- POS tagging
- Handwriting recognition

Figure: SuperTux Cart racing game [1].
**Problem Formulation**

**General notation:**
- \( \Pi \): Class of all possible policies, with \( \pi \in \Pi \) an arbitrary policy.
- \( T \): Task horizon, \( t \in T \) a specific time step.
- \( d^t_\pi \): Distribution of states in policy \( \pi \) at time step \( t \).
- \( d_\pi = \frac{1}{T} \sum_{t=1}^{T} d^t_\pi \): State distribution of policy \( \pi \) across all time steps.
- \( C(s, a) \): Immediate cost of an action \( a \) under a given state \( s \). Note that \( C \) is bound by \([0, 1]\).
- \( C_\pi(s) = \mathbb{E}_{a \sim \pi(s)}[C(s, a)] \): Expected immediate cost in state \( s \) under policy \( \pi \).
- \( J(\pi) = \sum_{t=1}^{T} \mathbb{E}_{s \sim d^t_\pi}[C_\pi(s)] = T \mathbb{E}_{s \sim d_\pi}[C_\pi(s)] \): Total cost of one episode under policy \( \pi \).
- \( \ell(s, \pi) \): Surrogate loss function (possibly with respect to an expert policy).
- \( Q_t^{\pi'}(s, \pi) \): \( t \)-step cost of executing \( \pi \) from the initial state \( s \) and then following \( \pi' \) after.
True cost of action $C(s, a)$ is usually unknown. Thus, we use the surrogate loss $\ell(s, \pi)$ instead.

Find a policy $\hat{\pi}$ that best approximates the expert policy $\pi^*$ under the distribution of states

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \mathbb{E}_{s \sim d_\pi} (\ell(s, \pi))$$

(1)

Shortcomings:

- Due to unknown system dynamics, cannot compute $d_\pi$.
  
  $\Rightarrow$ non-iid supervised learning problem, since representation of $d$ depends on $\pi$!
Outline

1. Motivation
2. Algorithm Descriptions
3. Analysis
4. Experimental Results
5. Limitations of the DAgger Algorithm
6. Conclusion
Reduction to Behavioral Cloning

Train classifier $D_{\text{sup}}$ only on states encountered by expert ($= d_{\pi^*}$), which yields policy $\pi_{\text{sup}}$:

$$\hat{\pi}_{\text{sup}} = \arg\min_{\pi \in \Pi} \mathbb{E}_{s \sim d_{\pi^*}} [\ell(s, \pi)]$$ (2)
Motivation  Algorithms  Analysis  Experiments  Limitations  Conclusion

Reduction to Behavioral Cloning

Train classifier $D_{sup}$ only on states encountered by expert ($= d_{\pi^*}$), which yields policy $\pi_{sup}$:

$$\hat{\pi}_{sup} = \arg\min_{\pi \in \Pi} \mathbb{E}_{s \sim d_{\pi^*}} [\ell(s, \pi)]$$  \hspace{1cm} (2)

Assume $\ell(s, \pi)$ is 0-1 loss, or upper bounded on 0-1 loss, implies:

**Theorem (2.1 Error of Behavioral Cloning)**

Let $\mathbb{E}_{s \sim d_{\pi^*}} [\ell(s, \pi)] = \epsilon$, then the resulting cost of the episode $J(\pi) \leq J(\pi^*) + T^2 \epsilon$.

For proof, see yesterday’s slides, or [2].
Iteratively trained policy
- Non-stationary
- $\pi_t$ for each time step $t$

$\pi_t$ trained to mimic $\pi^*$ on state distribution induced by previous policies $\pi_1, \ldots, \pi_{t-1}$

Thus guarantees expected loss to match average loss during training

Each policy is only adopted on its specific time step!
Theorem (2.1 Error of Behavioral Cloning)

Let $\pi$ be such that $\mathbb{E}_{s \sim d_\pi}[\ell(s, \pi)] = \epsilon$, and

$$Q^*_t(s, a) - Q^*_T(s, \pi^*) \leq u, \quad \forall a, t \in \{1, 2, \ldots, T\}, \quad d^*_t(s) > 0,$$

then it follows that $J(\pi) \leq J(\pi^*) + uT\epsilon$.

Also holds for any general policy $\pi$ that can guarantee $\epsilon$ surrogate loss!
Forward Training Guarantee

Proof: Consider $\pi$ that executes learned policy for first $t$ steps, then lets the expert policy $\pi^*$ take over. Then

$$J(\pi) = J(\pi^*) + \sum_{t=0}^{T-1} [J(\pi^*_{1:T-t}) - J(\pi^*_{1:T-t-1})] \text{ (deviation cost per time step)}$$  \hspace{1cm} (3)
Forward Training Guarantee

**Proof:** Consider $\pi$ that executes learned policy for first $t$ steps, then lets the expert policy $\pi^*$ take over. Then

$$J(\pi) = J(\pi^*) + \sum_{t=0}^{T-1} [J(\pi_{1:T-t}) - J(\pi_{1:T-t-1})] \quad \text{(deviation cost per time step)}$$  \hfill (3)

$$= J(\pi^*) + \sum_{t=1}^{T} \mathbb{E}_{s \sim d_T} [Q_{T-t+1}^*(s, \pi) - Q_{T-t+1}^*(s, \pi^*)] \quad \text{(per definition of } J(\pi))$$  \hfill (4)
Forward Training Guarantee

**Proof:** Consider \( \pi \) that executes learned policy for first \( t \) steps, then lets the expert policy \( \pi^* \) take over. Then

\[
J(\pi) = J(\pi^*) + \sum_{t=0}^{T-1} [J(\pi_{1:T-t}) - J(\pi_{1:T-t-1})] \quad \text{(deviation cost per time step)} \quad (3)
\]

\[
= J(\pi^*) + \sum_{t=1}^{T} E_{s \sim d_t}[Q^{\pi^*}_{T-t+1}(s, \pi) - Q^{\pi^*}_{T-t+1}(s, \pi^*)] \quad \text{(per definition of } J(\pi) \text{)} \quad (4)
\]

\[
\leq J(\pi^*) + u \sum_{t=1}^{T} E_{s \sim d_t}[\ell(s, \pi)] = J(\pi^*) + uT \epsilon \quad \text{(inequality from bounding on 0-1 loss)}.
\]

(5)
Stochastic Mixing Iterative Learning (SMILe) [2]

- Strongly related to SEARN
- Start from expert policy $\pi_0$
- At step $i$, $\hat{\pi}_i$ is trained to mimic expert under previous policy $\pi_{i-1}$

```
Initialize $\pi^0 \leftarrow \pi^*$ to query and execute expert.
for $i = 1$ to $N$ do
    Execute $\pi^{i-1}$ to get $D = \{(s, \pi^*(s))\}$.
    Train classifier $\hat{\pi}^i = \arg\min_{\pi \in \Pi} \mathbb{E}_{s \sim D}(e_\pi(s))$.
    $\pi^i = (1 - \alpha)^i \pi^* + \alpha \sum_{j=1}^{i} (1 - \alpha)^{j-1} \hat{\pi}^j$.
end for
Remove expert queries: $\tilde{\pi}^N = \frac{\pi^N - (1 - \alpha)^N \pi^*}{1 - (1 - \alpha)^N}$
```

Algorithm 4.1: The SMILe Algorithm.
Stochastic Mixing Iterative Learning (SMILe) [2]

- Update can be rewritten as $\pi_i = \pi_{i-1} + \alpha(1 - \alpha)^{i-1}(\hat{\pi}_i - \pi_0)$
- Generally $O(T^2)$ regret
- If parameter $\alpha \in O\left(\frac{1}{T^2}\right)$ guarantees near-linear regret in $T$ and $\epsilon$
- Also needs less iterations than SEARN ($O(T^2(\ln T)^{3/2})$ instead of $O(T^3 \ln T)$)
Choose arbitrary starting policy

Let policy $\hat{\pi}_i$ run, and flip coin whether $\hat{\pi}_i$ or expert $\pi^*$ execute current action

But always record expert decision (in the background)

Construct new dataset as aggregation of all previous samples

Train new policy, and repeat

---

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in $\Pi$.

$\textbf{for } i = 1 \textbf{ to } N \textbf{ do}$

Let $\pi_i = \beta_i \pi^* + (1 - \beta_i) \hat{\pi}_i$.
Sample $T$-step trajectories using $\pi_i$.
Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by $\pi_i$ and actions given by expert.
Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$.
Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.

$\textbf{end for}$

Return best $\hat{\pi}_i$ on validation.

**Algorithm 3.1: DAGGER Algorithm.**
Data Aggregation (DAgger)

- Avoid mixture of policies
- Follow-the-leader strategy avoids overfitting
- Mixture parameter $\beta_i$ generally indicator function $I(i = 1)$ or exponentially decaying value $p^{(i-1)}$

```
Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in $\Pi$.
for $i = 1$ to $N$ do
    Let $\pi_i = \beta_i \pi^* + (1 - \beta_i) \hat{\pi}_i$.
    Sample $T$-step trajectories using $\pi_i$.
    Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by $\pi_i$ and actions given by expert.
    Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$.
    Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.
```

**Algorithm 3.1: DAGGER Algorithm.**
1 Motivation

2 Algorithm Descriptions

3 Analysis

4 Experimental Results

5 Limitations of the DAgger Algorithm

6 Conclusion
Guarantee of Bounds to No-Regret Learning

- Assumes infinite sample trajectories at each iteration
- \( \epsilon_N = \min_{\pi} \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{s \sim d_{\pi_i}}[\ell(s, \pi)] \) true loss of best policy

**Theorem (3.1 Existence of Optimal Policy for Infinite Sample Case)**

For DAgger, if \( N \in \tilde{O}(T) \) there exists a policy \( \hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\} \) s.t.

\[
\mathbb{E}_{s \sim d_{\hat{\pi}}} [\ell(s, \hat{\pi})] \leq \epsilon_N + O\left(\frac{1}{T}\right).
\]

Proof via analysis results in next part.
Guarantee of Bounds to No-Regret Learning

- Holds for policy that performs best under its own distribution:
  \[ \hat{\pi} = \arg\min_{\pi \in \{\hat{\pi}_1, \ldots, \hat{\pi}_N\}} \mathbb{E}_{s \sim d_\pi} [\ell(s, \pi)] \]

- Alternatively, pick uniformly at random from \{\hat{\pi}_1, \ldots, \hat{\pi}_N\}
Guarantee of Bounds to No-Regret Learning

Combining Theorem 3.1 with Theorem 2.2 yields another result, important for the no-regret convergence. Only requirement is that $\ell$ upper bounds true cost $C$:

**Theorem (3.2 Convergence with respect to Expert Policy)**

For DAgger, if $N \in \tilde{O}(uT)$ there exists a policy $\hat{\pi} \in \{\hat{\pi}_1, \ldots, \hat{\pi}_N\}$ s.t.

$$J(\hat{\pi}) \leq J(\pi^*) + uT\epsilon_N + O(1).$$
Combining Theorem 3.1 with Theorem 2.2 yields another result, important for the no-regret convergence. Only requirement is that $\ell$ upper bounds true cost $C$:

**Theorem (3.2 Convergence with respect to Expert Policy)**

For DAgger, if $N \in \tilde{O}(uT)$ there exists a policy $\hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\}$ s.t.

$$J(\hat{\pi}) \leq J(\pi^*) + uT\epsilon_N + O(1).$$

**Proof:**

- 3.1 guarantees policy that satisfies prerequisites for 2.1
- Additional error bound of $O\left(\frac{1}{T}\right)$ over $T$ time steps is in $O(1)$. 
Guarantees for Finite Sample Case

- Usually only limited samples available
- Still guaranteed to find converging policy with certain probability
- Let $m$ be the samples per iteration
- Make use of a special case of the Azuma-Hoeffding inequality
Azuma-Hoeffding Inequality

- Gives probability that mean of samples from distribution are $\epsilon$-close to the actual mean of the distribution
- The more samples we have, the closer we can get

It states:

$$X_N = \sum_{i=1}^{N} Y_i, \quad \mathbb{E}[Y] = \mu$$

$$P[X_N/N - \mu > \epsilon/N] \leq e^{-\frac{\epsilon^2}{2N}} = \delta \quad (6)$$

$$\implies \epsilon = \sqrt{\log \frac{1}{\delta} \cdot 2N} \quad (7)$$

$$\implies P[X_N/N - \mu \leq \sqrt{\frac{2N \log \frac{1}{\delta}}{N}}] \geq 1 - \delta \quad (8)$$
Theorem (3.3 Convergence for Finite Sample Case)

For DAgger, if $N \in \tilde{O}(T^2 \log(1/\delta))$ and $m \in O(1)$, then with probability of at least $1 - \delta$ there exists a policy $\hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\}$ s.t. $\mathbb{E}_{s \sim d_{\hat{\pi}}} [\ell(s, \hat{\pi})] \leq \hat{\epsilon}_N + O\left(\frac{1}{T}\right)$. 

Theorem (3.4 Convergence for Finite Sample Case with respect to Expert)

For DAgger, if $N \in \tilde{O}(u^2 T^2 \log(1/\delta))$ and $m \in O(1)$, then with probability at least $1 - \delta$ there exists a policy $\hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\}$ s.t. $J(\hat{\pi}) \leq J(\pi^*) + u^T \hat{\epsilon}_N + O(1)$. 

Theorems for Finite Sample Case
Theorem (3.3 Convergence for Finite Sample Case)

For DAgger, if \( N \in \tilde{O}(T^2 \log(1/\delta)) \) and \( m \in O(1) \), then with probability of at least \( 1 - \delta \) there exists a policy \( \hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\} \) s.t.

\[
\mathbb{E}_{s \sim d_{\hat{\pi}}}[\ell(s, \hat{\pi})] \leq \hat{\epsilon}_N + O\left(\frac{1}{T}\right).
\]

Theorem (3.4 Convergence for Finite Sample Case with respect to Expert)

For DAgger, if \( N \in \tilde{O}(u^2 T^2 \log(1/\delta)) \) and \( m \) in \( O(1) \) then with probability at least \( 1 - \delta \) there exists a policy \( \hat{\pi} \in \{\hat{\pi}_1, ..., \hat{\pi}_N\} \) s.t.

\[
J(\hat{\pi}) \leq J(\pi^*) + u T \hat{\epsilon}_N + O(1).
\]
No-Regret Algorithms Guarantees

- Hold for any no-regret algorithm, not just Follow-the-leader
Further assumptions:

- $\beta_i$ is non-increasing
- $\ell_{\text{max}} \geq \ell_t(s, \hat{\pi}_t), \forall t \in \{1, \ldots, T\}$
- $n_\beta$ is largest $n$ s.t. $\beta_n > \frac{1}{T}$
Limitation of Average Regret

Further assumptions:

- $\beta_i$ is non-increasing
- $\ell_{\text{max}} \geq \ell_t(s, \hat{\pi}_t)$, $\forall t \in \{1, \ldots, T\}$
- $n_\beta$ is largest $n$ s.t. $\beta_n > \frac{1}{T}$

Lemma (4.1 Bound on Total Variation)

\[ \|d_{\pi_t} - d_{\hat{\pi}_t}\|_1 \leq 2T \beta_i, \text{ especially for } \beta_i \leq 1/T. \]
Limitation of Average Regret

Further assumptions:
- $\beta_i$ is non-increasing
- $\ell_{\text{max}} \geq \ell_t(s, \hat{\pi}_t), \forall t \in \{1, \ldots, T\}$
- $n_\beta$ is largest $n$ s.t. $\beta_n > \frac{1}{T}$

**Lemma (4.1 Bound on Total Variation)**
$$||d_{\pi_t} - d_{\hat{\pi}_t}||_1 \leq 2T\beta_i, \text{ especially for } \beta_i \leq \frac{1}{T}.$$  

**Theorem (4.1 Average Regret)**
For DAgger, there exists a policy $\hat{\pi} \in \{\hat{\pi}_1, \ldots, \hat{\pi}_N\}$ s.t.
$$\mathbb{E}_{s \sim d_{\hat{\pi}}} [\ell(s, \hat{\pi})] \leq \hat{\epsilon}_N + \gamma_N + \frac{2\ell_{\text{max}}}{N} [n_\beta + T \sum_{i=n_\beta+1}^N \beta_i], \text{ for } \gamma_N \text{ the average regret of } \{\hat{\pi}_1, \ldots, \hat{\pi}_N\}$$
Limitation of Average Regret

\[
\begin{align*}
\min_{\pi \in \Pi_{1:N}} \mathbb{E}_{s \sim d_{\pi}} [\ell(s, \hat{\pi})] \\
\leq \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{s \sim d_{\hat{\pi}_i}} (\ell(s, \hat{\pi}_i)) \\
\leq \frac{1}{N} \sum_{i=1}^{N} \left[ \mathbb{E}_{s \sim d_{\pi_i}} (\ell(s, \hat{\pi}_i)) + 2\ell_{\max} \min(1, T\beta_i) \right] \\
\leq \gamma_N + \frac{2\ell_{\max}}{N} [n_\beta + T \sum_{i=n_\beta+1}^{N} \beta_i] + \min_{\pi \in \Pi} \frac{1}{N} \sum_{i=1}^{N} \ell_i(\pi) \\
= \gamma_N + \epsilon_N + \frac{2\ell_{\max}}{N} [n_\beta + T \sum_{i=n_\beta+1}^{N} \beta_i]
\end{align*}
\]
Theorem (4.2 Average Regret in Finite Sampling Case)

For DAgger, with probability at least $1 - \delta$, there exists a policy $\hat{\pi} \in \{\hat{\pi}_1, \ldots, \hat{\pi}_N\}$ s.t.

$$\mathbb{E}_{s \sim d_{\hat{\pi}}} [\ell(s, \hat{\pi})] \leq \hat{\epsilon}_N + \gamma_N + \frac{2\ell_{\text{max}}}{N} \left[ n_\beta + T \sum_{i=n_\beta+1}^N \beta_i \right] + \ell_{\text{max}} \sqrt{\frac{2 \log(1/\delta)}{mN}},$$

for $\gamma_N$ the average regret of $\{\hat{\pi}_1, \ldots, \hat{\pi}_N\}$. 

Average Regret of Finite Sampling Case

\[
\begin{align*}
\min_{\hat{\pi} \in \hat{\pi}_{1:N}} \mathbb{E}_{s \sim d_{\hat{\pi}}} [\ell(s, \hat{\pi})] \\
\leq \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{s \sim d_{\hat{\pi}_i}} [\ell(s, \hat{\pi}_i)] \\
\leq \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{s \sim d_{\hat{\pi}_i}} [\ell(s, \hat{\pi}_i)] + \frac{2\ell_{\max}}{N} \left[ n_\beta + T \sum_{i=n_\beta+1}^{N} \beta_i \right] \\
= \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{s \sim D_i} [\ell(s, \hat{\pi}_i)] + \frac{1}{mN} \sum_{i=1}^{N} \sum_{j=1}^{m} Y_{ij} \\
+ \frac{2\ell_{\max}}{N} \left[ n_\beta + T \sum_{i=n_\beta+1}^{N} \beta_i \right] \\
\leq \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{s \sim D_i} [\ell(s, \hat{\pi}_i)] + \ell_{\max} \sqrt{\frac{2\log(1/\delta)}{mN}} \\
+ \frac{2\ell_{\max}}{N} \left[ n_\beta + T \sum_{i=n_\beta+1}^{N} \beta_i \right] \\
\leq \hat{\epsilon}_N + \gamma_N + \ell_{\max} \sqrt{\frac{2\log(1/\delta)}{mN}} + \frac{2\ell_{\max}}{N} \left[ n_\beta + T \sum_{i=n_\beta+1}^{N} \beta_i \right]
\end{align*}
\]
Comparison to SEARN [3]

**SEARN**
- Requires large number of iterations to converge
  - Both in theory and practice
- Mixture of policies
- [2] mention $O(T^2 \log T)$ instead of linear scaling in $T$ as presented in [3]

**DAgger**
- Stronger guarantees due to No-Regret approach
- Follow-the-leader returns single policy
- Requires less queries to expert (although still a lot)
Super Tux Cart

- Continuous action space (steering angle between [-1,1])

- DAgger performed best with $\beta_i = I(i = 1)$

Figure: Convergence of various methods for Super Tux Cart [1].
Super Mario Bros.

- Expert is near-optimal planning algorithm (expensive to query)
- Discrete action space (four buttons to press)
- Very simple levels!

**Figure:** Performance in Super Mario Bros. for various methods [1].
Handwriting Recognition

- Expert is supervised training data
- Discrete action space (predicted character)
- Probably sub-optimal compared to state-of-the-art neural architectures (RNN/LSTM)

Figure: Performance for handwriting recognition [1].
Outline

1. Motivation
2. Algorithm Descriptions
3. Analysis
4. Experimental Results
5. Limitations of the DAgger Algorithm
6. Conclusion
Limitations of DAgger

Main problem:

- DAgger still relies extremely heavily on the oracle/expert
- Each query can potentially be expensive, or make the collection of training samples hard
Limitations of DAgger

Main problem:
- DAgger still relies extremely heavily on the oracle/expert
- Each query can potentially be expensive, or make the collection of training samples hard
- Only guaranteed to work for convex loss functions
- Potentially a stronger bound can be given under the assumption of strong convexity
Limitations of DAgger

Main problem:

- DAgger still relies extremely heavily on the oracle/expert
- Each query can potentially be expensive, or make the collection of training samples hard
- Only guaranteed to work for convex loss functions
- Potentially a stronger bound can be given under the assumption of strong convexity
- Instability issues [4]
Limitations of DAgger

Main problem:

- DAgger still relies extremely heavily on the oracle/expert
- Each query can potentially be expensive, or make the collection of training samples hard
- Only guaranteed to work for convex loss functions
- Potentially a stronger bound can be given under the assumption of strong convexity
- Instability issues [4]

Still only at most performance on par with teacher!
Conclusion

- Presented various methods that are within $O(T^2)$ or lower bounds of an expert solution:
  - Forward Training
  - SMILe / SEARN
- Presented DAgger, a deterministic and stationary solution that alleviates several problems of previous methods by aggregating data across several episodes and querying an oracle/expert
- Provided extensive analysis of bounds for finite and infinite sample case for DAgger, which show nice properties
- Analyzed experiments, and showed some of DAgger’s limitations
Possible Extensions

Backplay curriculum learning [5, 6]:

- Instead of starting from initial starting state $s_0$, run the first iterations from an inverse policy $p_{t:T}$ that runs the expert for iterations $1:t$, and then the policy $\pi$.
- Potentially avoids distribution shift towards uncommon failure cases that appear in first iterations.
- Stabler training even with mixed policy for later episodes?
- Building a dense search tree from the bottom up could help to relinquish some queries: Instead of querying the expert, use surrogate loss as distance between expert replay and prediction (without expert).

Use with Deep Neural architectures:

- Experimental setup was conducted with linear SVM classifiers.
Stéphane Ross, Geoffrey Gordon, and Drew Bagnell.  
A reduction of imitation learning and structured prediction to no-regret online learning.  

Stéphane Ross and Drew Bagnell.  
Efficient reductions for imitation learning.  

Hal Daumé, John Langford, and Daniel Marcu.  
Search-based structured prediction.  
Hal Daumé.
A Course in Machine Learning.
http://ciml.info/.

Cinjon Resnick, Roberta Raileanu, Sanyam Kapoor, Alex Peysakhovich, Kyunghyun Cho, and Joan Bruna.
Backplay:” Man muss immer umkehren”.

Tim Salimans and Richard Chen.
Learning Montezuma’s Revenge from a single demonstration.
https://blog.openai.com/
learning-montezumas-revenge-from-a-single-demonstration/.
Thank you for your attention!