Introduction to DAgger

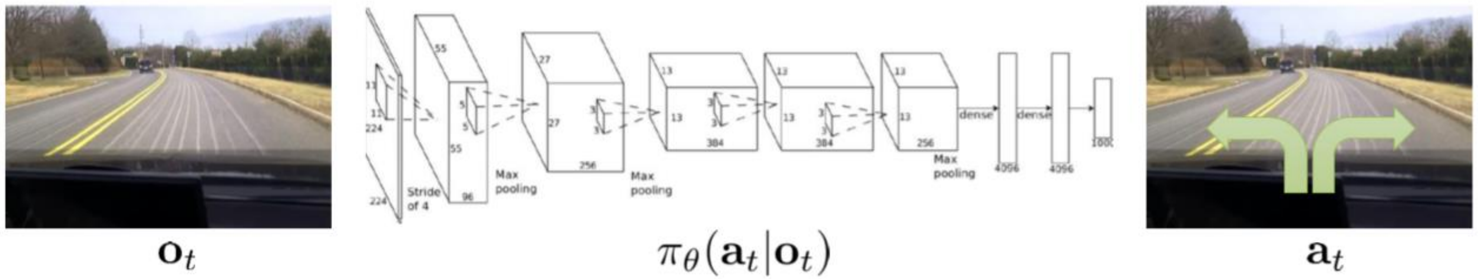
Shuijing Liu

CS 598 SG Paper Presentation

9/26/2019

Imitation learning

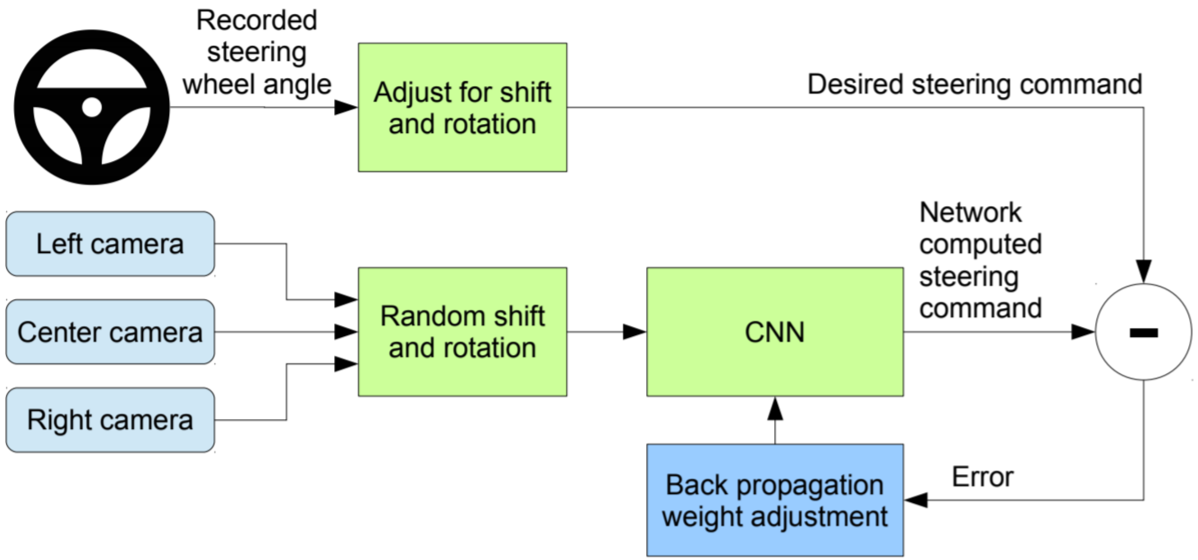
- Training a policy by imitating an expert’s behavior



http://rail.eecs.berkeley.edu/deeprlcourse-fa18/static/slides/lec-2.pdf

Imitation learning

- Nvidia Dave-2 neural network

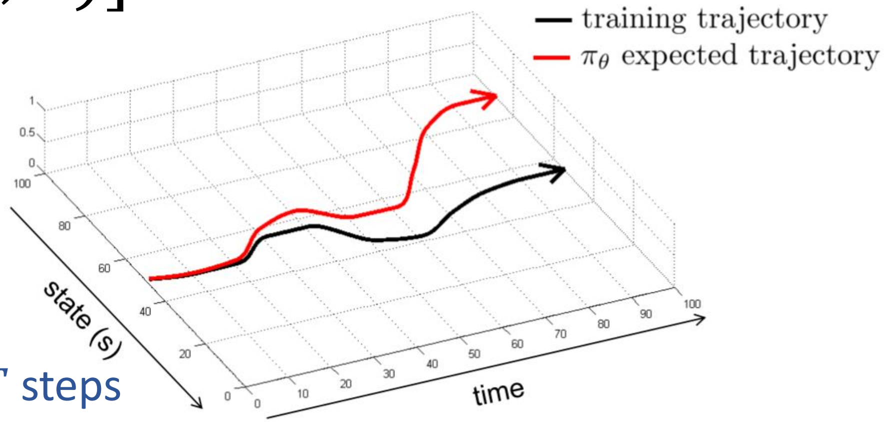


Bojarski, Mariusz, et al. "End to end learning for self-driving cars." *arXiv preprint arXiv:1604.07316* (2016).

Goal and supervised approach

- In imitation learning, our goal is to find a policy $\hat{\pi}$ which minimizes the surrogate loss ℓ under its induced distribution of states $d_{\hat{\pi}}$:
$$\hat{\pi} = argmin_{\pi \in \Pi} \mathbb{E}_{s \sim d_\pi}[\ell(s, \pi)]$$
- (**Supervised learning approach**) If we train a policy that learns to replicate π^* under the distribution of states encountered by the expert d_{π^*}:
$$\hat{\pi} = argmin_{\pi \in \Pi} \mathbb{E}_{s \sim d_{\pi^*}}[\ell(s, \pi)]$$

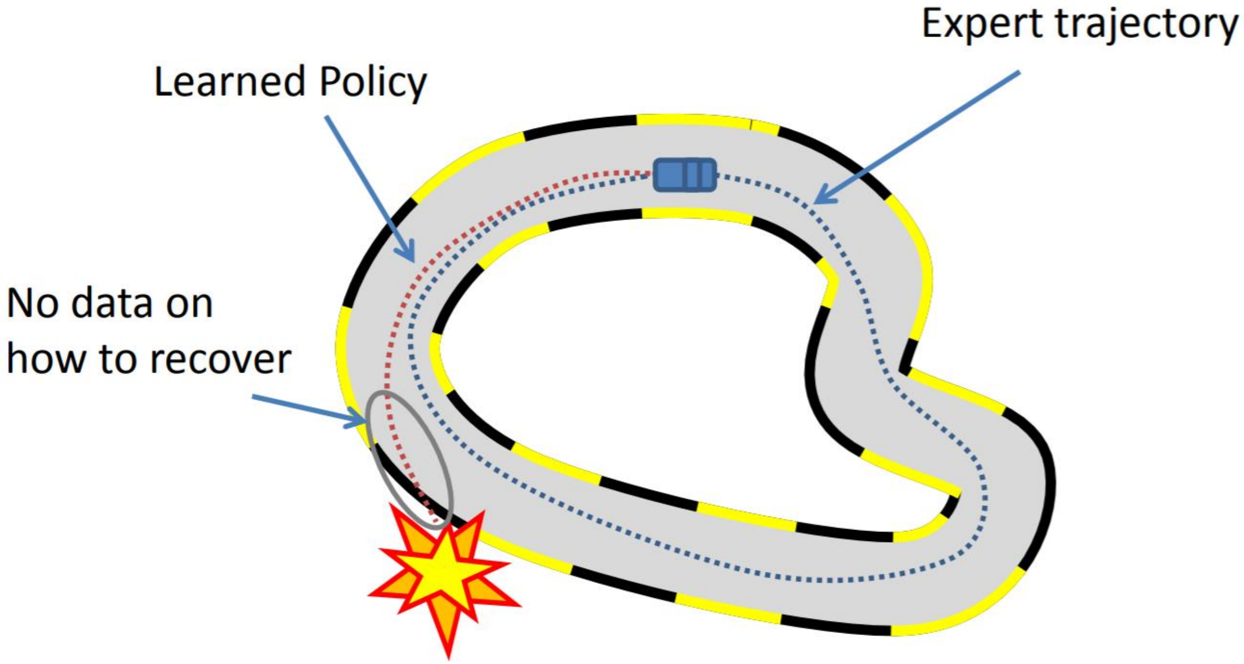
Will this work? No! (regret is quadratic)



Notations:
- Π: the class of all policies we consider
- π^*: expert policy
- d_π: $\frac{1}{T}\sum_{t=1}^{T} d_\pi^t$, the distribution of states we will visit if we follow policy π for T steps
- $\ell(s, \pi)$: the surrogate loss of π with respect to expert policy π^* in state s

The problem with supervised approach

- Data distribution mismatch



Forward training

- Instead, train T separate policies for each time step $t = 1, \ldots T$ and query the expert π^* under each policy's own state distribution d_π^t:

> Initialize $\pi_1^0, \ldots, \pi_T^0$ to query and execute π^*.
> **for** $i = 1$ **to** T **do**
>   Sample T-step trajectories by following π^{i-1}.
>   Get dataset $\mathcal{D} = \{(s_i, \pi^*(s_i))\}$ of states, actions taken by expert at step i.
>   Train classifier $\pi_i^i = \mathrm{argmin}_{\pi \in \Pi} \mathbb{E}_{s \sim \mathcal{D}}(e_\pi(s))$.
> $\pi_j^i = \pi_j^{i-1}$ for all $j \neq i$
> **end for**
> **Return** $\pi_1^T, \ldots, \pi_T^T$

So for every timestep t, we have a changing d_π^t instead of a single static d_π, which prevents the deviation of data distribution from d_{π^*}

- Forward training achieves near linear regret
- But forward algorithm is impractical for large T ☹

Stochastic mixture algorithms (SMILe and SEARN)

- At iteration n, the current policy π^n is a mixture of the old policy π^{n-1} and a new policy $\hat{\pi}^n$ trained by querying the expert π^* under $d_{\pi^{n-1}}$

$$\pi^n = (1-\alpha)\pi^{n-1} + \alpha\hat{\pi}^n$$

  where $\hat{\pi}^n = argmin_{\pi\in\Pi}\mathbb{E}_{s\sim d_{\pi^{n-1}}}[\ell(s,\pi)]$

- We can terminate after any iteration N, by removing the expert queries from our π^N and returning $\tilde{\pi}^N = \frac{\pi^N-(1-\alpha)^N\pi^*}{1-(1-\alpha)^N}$
- Regret is near linear

S. Ross and J. A. Bagnell. Efficient reductions for imitation learning. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), 2010.

Question: Can we do better?

Answer: Yes! Use DAgger.

DAgger algorithm

- DAgger trains a **deterministic** policy that achieves **no regret** in suitable conditions under **its induced distribution of states**

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

Notations:
- $\mathcal{D}$: dataset of state action pairs
- π^*: expert policy
- $\hat{\pi}$: policy trained to minimize
- π: a "mixture" policy that get executed at each iteration
- β_i: a decreasing coefficient s.t. $\frac{1}{N}\sum_{i=1}^{N} \beta_i \rightarrow 0$ as $N \rightarrow \infty$

DAgger algorithm

- In other words, at iteration i:
  - collect a trajectory $\{s_0, \ldots, s_T\}$ by rolling $\hat{\pi}_i$
  - Query the expert π^* for each state on the trajectory, to build a dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$
  - Train $\hat{\pi}_{i+1}$ with dataset aggregate
- Intuition: build a dataset that the final policy is likely to encounter based on previous experience
- Can be interpreted as a **Follow-the-Leader** algorithm, since it chooses the best next policy in hindsight

http://www.yisongyue.com/courses/cs159/lectures/imitation-learning-3.pdf

Question: Can we do even better than DAgger?

Answer: Yes, we can!

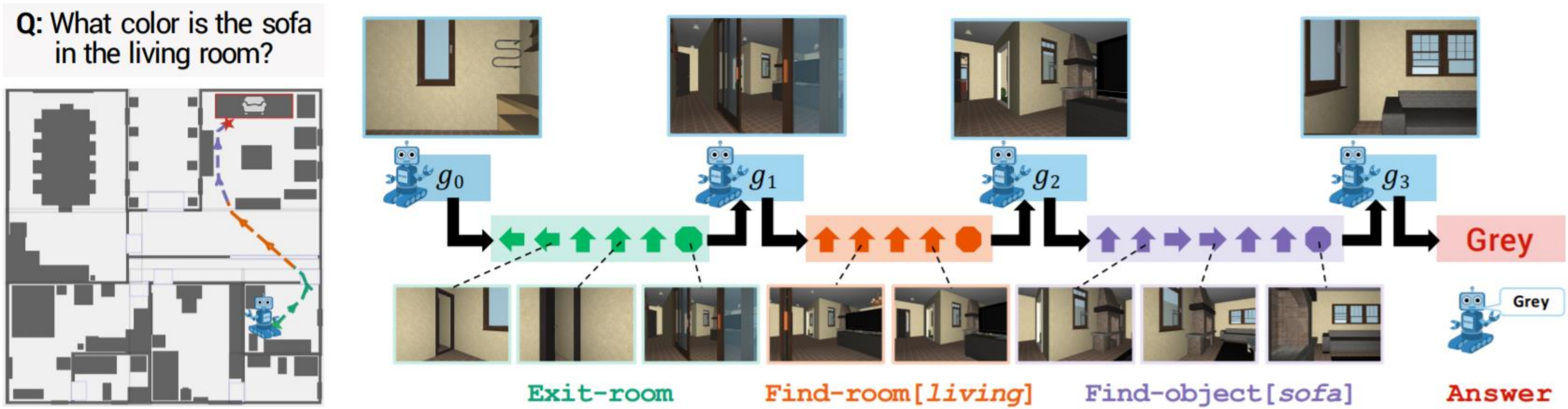
Extension of DAgger

- Problem: DAgger only cares about agreement with an expert, instead of the long term costs of various errors (For example, learning to drive near the edge of a cliff)
- **AGGREVATE**: learns to choose actions to minimize the cost-to-go of the expert, rather than the zero-one loss, $\ell(s, \pi)$, of mimicking its actions
- The performance boundaries of DAgger and AGGREVATE are identical, but AGGREVATE provides a stronger guarantee by bounding all losses by **regret** rather than by **error**

Ross, Stephane, and J. Andrew Bagnell. "Reinforcement and imitation learning via interactive no-regret learning." *arXiv preprint arXiv:1406.5979* (2014).

Applications

- Embodied Question Answering (https://embodiedqa.org/)



A. Das, G. Gkioxari, S. Lee, D. Parikh, D. Batra; 2nd Annual Conference on Robot Learning, CoRL 2018, 2018, pp. 53-62

Problems and Discussions

- DAgger and other IL algorithms need data from human, which is finite and expensive
  - Deep learning works best when data is plentiful
- Can they do better than the human expert?
  - Humans are not good at providing some kinds to actions
  - Combine of IL and RL?

Thank you!