1. Policy Evaluation
(Online Bellman Residual)
[Sun & Bagnell, 15, UAI (Best Student Paper)]
Function Approximation

2. RL via Imitation
(Imitation Learning)
[Sun et.al 17, ICML; 18, ICLR]
Function Approximation & Imitation

3. RL via Indirect Imitation
(Dual Policy Iteration)
[Sun et.al, 18, submitted to ICML]
Function Approximation
Optimal Control

4. Proposed Work:
Temporal Difference Learning & Apprenticeship Learning
Imitation Learning

- SVM
- Gaussian Process
- Kernel Estimator
- Deep Networks
- Random Forests
- LWR
- ...

Maps states to actions
Efficient RL via Imitation

Extra Assumptions:
- Expert policy $\pi^e$ at training;
- Cost (reward) signal (e.g., Optimal Planner, MPC);
- [Choudhury, et.al, ICRA, RSS, 17, Pan et.al, 17]

Human

Planner & Controller (robotics)

Ground Truth Labels + Utility (NLP)

Search Algorithm (e.g. A*) as expert
Why bother imitating when you have cost/reward signals?
Formalizing Advantages

ADVANTAGE #1
Less sensitive to local optimality

DAgger (Data Aggregation) [Ross et.al, 11, AISTATS]
AggreVaTe (Aggregate with Values) [Ross&Bagnell14, arxiv]

ADVANTAGE #2
More Sample Efficient (i.e., Learns faster)

There exist problems, s.t. with access to optimal expert, i.e., $\pi^e = \pi^*$

IL learns exponentially faster than RL

[Sun et.al, 17,ICML]
Consider a simple, tree like MDP

let us assume $\pi^e = \pi^*$, and we can query $Q^*(s, a)$

$$Q^\pi(s, a) : \text{the total cost of taking action } a \text{ at } s \text{ then following policy } \pi$$
And an expert who tells us which action is better

let us assume \( \pi^e = \pi^* \), and we can query \( Q^*(s, a) \)

\[ Q^*(s_0, a_l) < Q^*(s_0, a_r) \]

\[ Q^*(s_1, a_l) < Q^*(s_1, a_r) \]

Halving: Eliminate half of the nodes at every iteration

\[ c_3 < c_4 < c_5 < c_6 \]
In time logarithmic in \#states, we know an optimal policy
In time logarithmic in #states, we know an optimal policy.

Now if we can only query unbiased but noisy $Q^*$

\[
\sum_{i=1}^{N} (J(\pi_i) - J(\pi^*)) \leq O\left( \ln(S)\left(\sqrt{\ln(S)N} + \sqrt{\ln(2/\delta)N}\right)\right)
\]

Poly-Log wrt $S$
But For Pure RL

\[ s_0 \]

\[ s_1 \]

\[ s_3 \quad s_4 \]

\[ s_5 \quad s_6 \]

\[
\begin{align*}
c_3 &< c_4 < c_5 < c_6 \\
\sum_{i=1}^{N} (J(\pi_i) - J(\pi^*)) &\geq \Omega(\sqrt{SN})
\end{align*}
\]

ANY RL ALGORITHM: Proof uses a reduction from Multi-Armed Bandit

[Sun et.al, 17, ICML]
Roll in Learned Policy, Stop at a randomly picked time step

\[ Q^*(s, a_1) = 100 \]

\[ Q^*(s, a_2) = 0 \]

\[ Q^*(s, a_2) = 3 \]

Cost-Sensitive classification dataset

\[ \{ s, \begin{bmatrix} Q^*(s, a_1) \\ Q^*(s, a_2) \\ Q^*(s, a_3) \end{bmatrix} \}^N \]
Ex: AggreVaTe

Cost-Sensitive Classifier

\[ \pi_{n+1} = \arg \min_{\pi} \sum_{s \in \mathcal{D}} \sum_{a} \pi(a|s)Q^*(s,a) \]

Aggregate Dataset

\[ \mathcal{D} + \{s, \begin{bmatrix} Q^*(s,a_1) \\ \vdots \\ Q^*(s,a_A) \end{bmatrix} \}^N \]

Simple incremental update?
Towards Differentiable AggreVaTe (AggreVaTeD)

[Sun et.al, 17, ICML]

Stochastic parameterized policy:

$$\pi_\theta = \pi(\cdot|s; \theta)$$

$\pi(\cdot|s) \in \Delta(A)$

Non-Linear Layer → Softmax Layer
Differentiable AggreVaTe (AggreVaTeD)

\[ \nabla_\theta \ell_n(\theta) \mid_{\theta_n} \quad \ell_n(\theta) = \sum_s \sum_a \pi(a \mid s; \theta) Q^*(s, a) \]

**AggreVaTeD-GD (Gradient Descent):**
\[ \theta_{n+1} = \theta_n - \mu \nabla_\theta \ell_n(\theta) \mid_{\theta=\theta_n} \]

**AggreVaTeD-NG (Natural Gradient):**
\[ \theta_{n+1} = \theta_n - \eta_n I(\theta_n)^{-1} \nabla_\theta \ell_n(\theta_n) \]

Cost-Sensitive loss on the new batch (no aggregation)
Differentiable AggreVaTe (AggreVaTeD)

Discrete MDP, Tabular Policy:

- AggreVaTeD-GD
- AggreVaTeD-NG

AggreVaTe with Online Gradient Descent
[Zinkevich, 03]

AggreVaTe with Weighted Majority
[Littlestone & Warmuth, 03]

- Practical Algorithms
- AggreVaTe’s theory as Guidance
- GD ensures Convergence

Strong Theoretical Guarantee

\[ J(\hat{\pi}) \leq J(\pi^e) \]
Dependency Parsing

Dependency Parsing on Handwritten Algebra Data

\[-5(x-1) = -20\]

\[x - 1 = 4\]

\[x = 5\]
Dependency Parsing

Dependency Parsing <=> Sequential Decision Making

[Chang et.al 15, Duyck & Gordon 15]

Partial constructed parse tree + Action (Arc) => New partial parse tree

\[ S_t + a_t \Rightarrow S_{t+1} \]

36
Performance of AggreVaTeD, RL, and DAgger

Unlabelled Attachment Score

The Higher the Better

AggreVaTeD (LSTM) 82.6 53 38.6 83.2
RL (LSTM) 92.4
AggreVaTeD (NN) 53
RL (NN) 38.6
DAgger 83.2

RL: Natural Policy Gradient [Kakade02, Bagnell 04]
DAgger result from Duyck & Gordon, 15
“Surprisingly”
It Can Outperform Expert

CartPole and Acrobot

The Higher the Better

R (y-axis): total reward

Experts