Sequential Tasks

• Versus pure Learning from Demonstration, we seek to:
  • Minimize uncertainty
  • Maximize smoothness

• Sequential Task
  • Implicitly or Explicitly looks ahead
  • Goal 1: Do what human wants you to do
  • Goal 2: Outperform human and outperform autonomous learning
Section Outline

• Autonomous Learning: RL
• Demonstration + RL
  • action selection
  • shaping reward
  • IRL: shaping reward
• Learning from human feedback
  • Treat as environment reward
  • Treat as return
  • Return + RL
  • Treat as categorical feedback regrading policy
Reinforcement Learning (RL) Goals

- TD-Gammon beat Professionals: Tesauro, 1995
- Aibo Learned More Effective Gait: Kohl & Stone, 2004
- AlphaGo achieved Super-human performance: Silver et al., 2016

Learning autonomously is often better than hand-coding!

But not always!
**RL Setting**

**Markov Decision Process (MDP)**

- $S$: set of states in the world
- $A$: set of actions an agent can perform
- $T$: $S \times A \rightarrow S$ (transition function)
- $R$: $S \rightarrow \mathbb{R}$ (environmental reward)
- $\pi$: $S \rightarrow A$ (policy)
- $Q$: $S \times A \rightarrow \mathbb{R}$ (action-value function)
RL References

• Sutton & Barto, “Reinforcement Learning: An Introduction”

• Littman & Isbell Udacity course, “Reinforcement Learning”
  https://classroom.udacity.com/courses/ud600/

• Szepesvári, “Algorithms for Reinforcement Learning”
  https://sites.ualberta.ca/~szepesva/RLBook.html

• (Many others too)
RL & Speed

• Need data to learn. Can be equivalent to time
• Often start with random bias
• RL is worst at beginning (by definition?)
• Many techniques to achieve better Bias
  • Transfer Learning
  • Constrained action/state space
  • Hand-coded generalization

• Today: Bias from a human
Section Outline

• Autonomous Learning: RL
• Demonstration + RL
  • action selection
  • shaping reward
  • IRL: shaping reward
• Learning from human feedback
  • Treat as environment reward
  • Treat as return
  • Return + RL
  • Treat as categorical feedback regrading policy
HAT: Human-Agent Transfer

1. Observe Human Demonstration (or suboptimal controller)
2. Summarize Policy
3. Bootstrap Autonomous Learning with summarized policy

Hold Ball, Pass₁, Pass₂

“Integrating Reinforcement Learning with Human Demonstrations of Varying Ability”. Taylor, Suay, & Chernova, 2011
1. Human Demonstration
2. Summarize Policy
3. Autonomous Learning

In state s, evaluate agent’s 3 actions
\[ Q(s, a_1) = 5 \]
\[ Q(s, a_2) = 3 \]
\[ Q(s, a_3) = 4 \]

And evaluate action suggested by decision list
\[ D_{target}(s) = a_3 \]

- P(Execute): Take D(s) action
- P(Explore): Take random action
- P(Exploit): Take action w/ max Q

P(Execute) = 1: tries to mimic human
HAT: Human-Agent Transfer Results

Improvements with only ~3 minutes of human time

Example Policy
HAT: Human-Agent Transfer

**Initiation:** Student
**Modality:** Trajectories
**Live/Offline:** Offline
**Present/Remote:** N/A
**Expertise:** Any
**Investment:** N/A
**Learning Paradigm:** RL
**Data Sources:** Human provided + environment provided
**Individual/Team Goal:** Learner acts alone
**Training/Testing:** Tested on training task
Confidence HAT

• Goal: Improve Reinforcement Learning with Confidence-Based Demonstrations

“Improving Reinforcement Learning with Confidence-Based Demonstrations”. Wang & Taylor, IJCAI 2017
Confidence HAT

• Source demonstration quality?
• Source demonstration consistency?
• Summarization quality?
• Task coverage?

“Improving Reinforcement Learning with Confidence-Based Demonstrations”. Wang & Taylor, 2017
Confidence HAT

• 3-step method:

  • Source Demonstration → Train → Confidence Model → Transfer → Target Agent Advice

• Uncertainty measurement of demonstration
  • Summarize demonstration data into confidence-based models
  • Provide action suggestions along with confidence: let target agent decide (threshold)
  • Integrate with RL, help improve initial learning and overall performance

“Improving Reinforcement Learning with Confidence-Based Demonstrations”. Wang & Taylor, 2017
Where does the confidence model come from?

Summarize the prior knowledge into Gaussian Process model

\[ P(\omega_i|x) = \frac{1}{\sqrt{2\pi|\Sigma_i|}} \exp\left\{ -\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right\} \]

- Confidence Function
Keepaway Domain

Demonstration:
• State-action pairs of 20 episodes

GPHAT:
• Cluster active data (Pass1 & Pass2) into smaller groups.
• Train Gaussian classifiers upon smaller clusters.
• Set a threshold. Follow GPHAT’s suggested action with confidence higher than that. (e.g. 0.9 is a reasonable value).

Probabilistic policy reuse:
• Prior knowledge would be reused with a decaying probability
Performance Improvement

Mario domain
DAgger (Dataset Aggregation)

- Iterative algorithm
- Trains a stationary deterministic policy
- No regret algorithm in an online learning setting

[under reasonable assumptions, it] “must find a policy with good performance under the distribution of observations it induces in such sequential settings”

“A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning”. Ross, Gordon, & Bagnell, 2011
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.

At the first iteration, it uses the expert's policy to gather a dataset of trajectories $\mathcal{D}$ and train a policy $\hat{\pi}_2$ that best mimics the expert on those trajectories. Then at iteration $n$, it uses $\hat{\pi}_n$ to collect more trajectories and adds those trajectories to the dataset $\mathcal{D}$. The next policy $\hat{\pi}_{n+1}$ is the policy that best mimics the expert on the whole dataset $\mathcal{D}$.

**Insight:**
1) Combine learned policy with novel human demos
2) Train over all of human demos
3) Learn about areas of the state space not initially reached
Super Tux Kart
AggreVaTe (Aggregate Values to Imitate)

- Expected future cost-to-go: $Q^\pi_t(s, a)$ of executing $a$ in $s$, and then following $\pi$ for $t-1$ steps
- $d^t_\pi$ distribution of states at time $t$ induced by executing policy $\pi$
- Overall performance: $J(\pi) = \sum_{t=1}^{T} \mathbb{E}_{s \sim d^t_\pi}[C(s, \pi(s))]$
- Observe expert perform task
- At uniformly random time, explores an action $a$ in state $s$, and then get cost-to-go $Q$ after performing this action
- Choose actions to minimize co-to-go instead of classification loss

"Reinforcement and Imitation Learning via Interactive No-Regret Learning". Ross & Bagnell, 2014
Algorithm 1 AGGREGVATE: Imitation Learning with Cost-To-Go

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

for $i = 1$ to $N$ do

Let $\pi_i = \beta_i \pi^* + (1 - \beta_i) \hat{\pi}_i$ #Optionally mix in expert’s own behavior.

Collect $m$ data points as follows:

for $j = 1$ to $m$ do

Sample uniformly $t \in \{1, 2, \ldots, T\}$.

Start new trajectory in some initial state drawn from initial state distribution.

Execute current policy $\pi_i$ up to time $t - 1$.

Execute some exploration action $a_t$ in current state $s_t$ at time $t$.

Execute expert from time $t + 1$ to $T$, and observe estimate of cost-to-go $\hat{Q}$ starting at time $t$.

end for

Get dataset $D_i = \{(s, t, a, \hat{Q})\}$ of states, times, actions, with expert’s cost-to-go.

Aggregate datasets: $D \leftarrow D \cup D_i$.

Train cost-sensitive classifier $\hat{\pi}_{i+1}$ on $D$.

(Alternately: use any online learner on the data-sets $D_i$ in sequence to get $\hat{\pi}_{i+1}$)

end for

Return best $\hat{\pi}_i$ on validation.
• Task performance of learned policies: related to regret on regression loss and the cost-to-go

• Task performance relates to the square root of the online learning regret and the regression regret of the best regressor in the class to the Bayes-optimal regressor on this training data

• Potential drawback: “any method relying on cost-to-go estimates can be impractical as collecting each estimate for a single state-action pair may involve executing an entire trajectory”
LfD + Shaping Rewards: Similarity Based Shaping

• RL + LfD: RLFD

• Want high potential function when action was demonstrated nearby

• Given demonstrations & similarity/distance function:
  • Create potential shaping function on the fly

• Think: placing Gaussians on demonstrated (s,a)
  • Local reward shaping

LfD + Shaping Rewards: Similarity Based Shaping

- Define similarity measure between states and actions

\[
\text{sim}(s, a, s^d, a^d, \Sigma) = \begin{cases} 
0 & \text{if } a \neq a^d \\
\exp\left(-\frac{1}{2}(s-s^d)^T\Sigma^{-1}(s-s^d)\right) & \text{if } a = a^d
\end{cases}
\]

- Set potential to highest similarity among demonstrated samples

\[
\Phi(s, a) = \max_{(s^d, a^d)} \text{sim}(s, a, s^d, a^d, \Sigma)
\]
- RL ($Q(\lambda)$-learning)
- RLfD ($Q(\lambda)$-learning+shaping)
- RLfD ($Q(\lambda)$-learning+HAT)
- LfD (C4.5 decision tree classifier [Quinlan, 1993])

Learning on Mario from 1 demonstration
Inverse Reinforcement Learning

MDP/R


Model-free IRL:

IRL + Shaping: Static IRL Shaping (SIS)

• Collect demonstrations: \((s_1, a_1, s_2, s_2, \ldots)\)
• Learn reward function over states using IRL
• Use new reward function as potential-based shaping reward over states:
  \[ F(s, a, s') = \gamma \Phi(s') - \Phi(s) \]
  \[ R' = R + F \]
• Potential function does not change over time
• The effect of shaping is that the agent’s exploration is less random and the agent is biased towards states with high potential

“Learning from Demonstration for Shaping through Inverse Reinforcement Learning”. Suay, Brys, Taylor, & Chernova, 2016
Collect demonstrations: \((s_1,a_1,s_2,a_2,...)\)

Learn reward function using states and actions IRL

Use dynamic shaping: 
\[
F(s,a,t,s',a',t') = \gamma \Phi(s',a',t') - \Phi(s,a,t)
\]

Learn secondary Q-function online for potential function

- \(\Phi_2(s,a) \leftarrow \Phi_2(s,a) + \alpha_2(r_{\text{IRL}}(s) + \gamma \Phi_2(s',a') - \Phi_2(s,a))\)
- Q-function gets updated online after each observation

Now use this (changing) potential-based function:

- \(F = \gamma \Phi_2(s',a') - \Phi_2(s,a)\)
- \(R' = R + F\)
• Autonomous Learning: RL
• Demonstration + RL
  • action selection (time to go)
  • shaping reward
  • IRL: shaping reward
• Learning from human feedback
  • Treat as environment reward
  • Treat as return
  • Return + RL
  • Treat as categorical feedback regrading policy
Learning *Directly* from Human Reward

- Sophie’s Kitchen
- Human trainer can award a scalar reward signal $r = [-1, 1]$

```
Algorithm 1 Q-Learning with Interactive Rewards:
s = last state, \( s' \) = current state, \( a \) = last action, \( r \) = reward

1: while learning do
2: \hspace{1em} \( a \) = random select weighted by \( Q[s, a] \) values
3: \hspace{1em} execute \( a \), and transition to \( s' \)
   \hspace{1em} (small delay to allow for human reward)
4: \hspace{1em} sense reward, \( r \)
5: \hspace{1em} update values:
   \hspace{1em} \( Q[s, a] \leftarrow Q[s, a] + \alpha (r + \gamma \max_{a'} Q[s', a'] - Q[s, a]) \)
6: end while
```

Learning *Directly* from Human Reward

• Anticipatory Guidance Rewards

• “Even though our instructions clearly stated that communication of both general and object specific rewards, we found many people assumed that object specific rewards were future directed messages or guidance for the agent. Several people mentioned this in the interview, and we also find behavioral evidence in the game logs.”

• They provide
  1) anticipatory reward (direct future) &
  2) feedback for past actions
Algorithm 2 Interactive Q-Learning modified to incorporate interactive human guidance in addition to feedback.

1: while learning do
2:   while waiting for guidance do
3:     if receive human guidance message then
4:       \( g = \text{guide-object} \)
5:     end if
6:   end while
7:   if received guidance then
8:     \( a = \text{random selection of actions containing } g \)
9:   else
10:    \( a = \text{random selection weighted by } Q[s,a] \text{ values} \)
11: end if
12: execute \( a \), and transition to \( s' \)
13: (small delay to allow for human reward)
14: sense reward, \( r \)
15: update values:
16: \[
Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma(\max_{a'} Q[s',a']) - Q[s,a])
\]
17: end while
Learning from Human Rewards: Interactive Shaping

Key insight: trainer evaluates behavior using model of its long-term quality

Learn a model of human reinforcement

\[ H : S \times A \rightarrow \mathbb{R} \]

Directly exploit the model to determine action

Also, can combine with MDP’s reward

http://www.cincinnatireview.com/blog/tag/lion-tamer/

http://www.bradknox.net/projects/
TAMER Learning Tetris

Initial Training

After 2 games of Training
TAMER+RL

- 2 settings
  - Sequential
  - Simultaneous

- Important points:
  - Decaying influence
  - Eligibility traces for reward


Rewards shaping: \( R'(s, a) = R(s, a) + (\beta \ast \hat{H}(s, a)) \)

Q augmentation: \( Q'(s, a) = Q(s, a) + (\beta \ast \hat{H}(s, a)) \)

Action biasing: \( Q'(s, a) = Q(s, a) + (\beta \ast \hat{H}(s, a)) \) only during action selection

Control sharing: \( P(a=\arg\max_a [\hat{H}(s, a)]) = \min(\beta, 1) \). Otherwise use base RL agent’s action selection mechanism.
Motivation: Dog Training

• Teach dog to sit & shake

Policy

• Mapping from observations to actions
• Feedback: \{Bad Dog, Good Boy\}
History of Evidence

- Feedback history $h$
- ...
- Really make sense to assign numeric rewards to these?
Bayesian Framework

• Trainer desires policy \( \lambda^* \)
• \( h_t \) is the training history at time \( t \)
• Find MAP hypothesis of \( \lambda^* \):

\[
\arg\max_{\lambda} p(\lambda^* = \lambda | h_t) = \arg\max_{\lambda} p(h_t | \lambda^* = \lambda) p(\lambda^* = \lambda)
\]

Prior distribution over policies

Model of training process

Strategy-Aware Bayesian Learning (SABL)

Assuming trainer feedback is given according to a probabilistic model (with known $\mu^+$, $\mu^-$ and $\varepsilon$)

- action was correct, with error probability $\varepsilon$
- withhold or give explicit feedback, with probability $\mu^+$ and $\mu^-$

Compute a maximum likelihood estimate of the target policy $\lambda$, given a training history $h$:

$$\lambda^* = \arg\max_{\lambda} Pr[h|\lambda, \mu^+, \mu^-, \varepsilon]$$
To a strategy-aware learner, the lack of feedback can be as informative as explicit feedback.

No feedback?

That is not what I want, try something else!

Keep going and you will get reward eventually!
Infer Neutral

- Try to learn what no-reward ($\mu^+ \& \mu^-$) means
- Don’t assume they’re balanced

- Many trainers don’t use punishment
  - Neutral feedback = punishment

- Some don’t use reward
  - Neutral feedback = reward
How Humans Reward

- Turkers & Dog Training Enthusiasts
- Explicitly reward good behavior? R+
- Explicitly punish bad behavior? P+
- Stay consistent over time?

Protect the Field

<table>
<thead>
<tr>
<th></th>
<th>(P_+)</th>
<th>(P_-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R_+)</td>
<td>93</td>
<td>125</td>
</tr>
<tr>
<td>(R_-)</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>begin</th>
<th>end (R_+/P_+)</th>
<th>(R_+/P_-)</th>
<th>(R_-/P_+)</th>
<th>(R_-/P_-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R_+/P_+)</td>
<td>65</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>(R_+/P_-)</td>
<td>10</td>
<td>52</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(R_-/P_+)</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>(R_-/P_-)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
How Humans Reward

- Get the battery
- Eat the bird
- Point towards the box

Protect the Field
Policy Shaping

• Simulated Oracle: theoretical analysis

• Combines human feedback with RL

• Positive and negative trainer feedback = discrete communication that depends on trainer’s target policy

• Feedback can be correct [consistent] with some probability $C$ and human will provide feedback with some likelihood $L$

Difference between number of “right” and “wrong” labels: $\Delta_{s,a}$

Prob $s,a$ is optimal (binomial distribution):

$$\frac{C^{\Delta_{s,a}}}{C^{\Delta_{s,a}} + (1 - C)^{\Delta_{s,a}}}$$

Combine probabilities of different actions based on learned Q-values (Bayesian Q-Learning) and critique advice

$$\frac{P_q(a)P_c(a)}{\sum_{a \in A} P_q(a)P_c(a)}$$

Very similar to Q-learning when
1. Small amount of human critique
2. Critique equal among many $s/a$ pairs
3. Human is right roughly half the time
   (C is close to 0.5)
Policy Shaping

• 2nd paper: focus on human participants

• Participants: shown videos of recorded trajectories

• Goals:
  • Humans vs. Oracle
  • Value of silence

• Provide positive or negative feedback

• Error rate and assumptions re: +/- set by fixed params

Policy Shaping

Investigate:
- Humans can provide good data for shaping
- People have inherent bias regarding silence
- Can manipulate meaning of silence

Experiments
- Oracle: simulated teacher
- Human-unbiased: a human teacher provides action critiques, with no instruction about the meaning of silence.
- Human-positive bias: instruction that silence is positive
- Human-negative bias: with instruction that silence is negative
Primary result: Humans could give useful feedback

• “Even when giving instructions biasing silence towards bad, it is still better to assume that silence means good.”
• “It could be that people tend to mean silence as good”
• “However, to fully convince ourselves of this we would need to experiment on a variety of domains with different positive/ negative biases”
Aside: Learning lower-level skills

• (e.g., Dynamic Motion Primitives)
• Particularly important in Robotics
Open Questions: 1/2

• Two-way communication
  • Asking for help
  • Human knows what robot knows
    • Robot knows human knows what robot knows
      • Human knows robot knows human knows what robot knows…

• Steer human towards useful feedback
  • Reciprocal interaction
  • Human effective at shaping a given agent.
  • “Eliciting good teaching from humans for machine learners”. Cakmak & Thomaz, 2014

• Curriculum Learning
Open Questions: 2/2

• Best way to teach people to teach?
• Different modalities
  • LfD vs. LfF
• Treating experts of different quality differently
• Testing with *normal people*
• Crowdsourcing?
References

• Littman & Isbell Udacity course, “Reinforcement Learning” https://classroom.udacity.com/courses/ud600/
• “Integrating Reinforcement Learning with Human Demonstrations of Varying Ability”. Taylor, Suay, & Chernova, 2011
• “Improving Reinforcement Learning with Confidence-Based Demonstrations”. Wang & Taylor, 2017
• “A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning”. Ross, Gordon, & Bagnell, 2011
• “Reinforcement and Imitation Learning via Interactive No-Regret Learning”. Ross & Bagnell, 2014
• “Reinforcement Learning from Demonstration through Shaping”. Brys, Harutyunyan, Suay, Chernova, Taylor, and Nowé, 2015.
• Relative Entropy Inverse Reinforcement Learning”. Boularias, Kober, & Peters, 2011.
• “Learning from Demonstration for Shaping through Inverse Reinforcement Learning”. Suay, Brys, Taylor, & Chernova, 2016
• “Reinforcement Learning from Simultaneous Human and MDP Reward”. Knox & Stone, 2012.
• “Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning”. Loftin, Peng, MacGlashan, Littman, Taylor, Huang, & Roberts, 2015
• “Policy shaping: Integrating human feedback with reinforcement learning”. Griffith, Subramanian, Scholz, Isbell, & Thomaz, 2013