

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Inverse Reinforcement Learning

Many slides courtesy of Dan Klein, Stuart Russell, Andrew Moore, or Rose Yu

CS 5300 / CS 6300
Artificial Intelligence
Spring 2009

Hal Daumé III
hal@cs.utah.edu

www.cs.utah.edu/~hal/courses/2009S_AI

Slide 1

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Announcements

- Project 1 extra credit demos and winners!
- P3 out
 - (but a few changes later today/tomorrow)
- Competition out next week
 - Details from last year at: <http://inst.eecs.berkeley.edu/~cs188/fa08/projects/contest/contest.html>
 - Two agents per team (red and blue)
 - Try to capture all of opponents dots (no other points)
 - Can only see 5 squares away (otherwise, noisy reading)
 - Game ends after time limit or all dots eaten
 - 0.5 seconds per agent per move
 - Teams may be of any size

Slide 2

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

1. Algorithms for Inverse Reinforcement Learning

2. Apprenticeship learning via Inverse Reinforcement Learning

Slide 4

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Inverse RL: Motivation

- Given: (1) measurements of an agent's behavior over time, in a variety of circumstances, (2) if needed, measurements of the sensory inputs to that agent; (3) if available, a model of the environment.
- Determine: the reward function being optimized.

Slide 5

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Why?

- Reason #1: Computational models for animal and human learning.
- "In examining animal and human behavior we must consider the reward function as an unknown to be ascertained through empirical investigation."
- Particularly true of multiattribute reward functions (e.g. Bee foraging: amount of nectar vs. flight time vs. risk from wind/predators)

Slide 6

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Why?

- Reason #2: Agent construction.
- "An agent designer [...] may only have a very rough idea of the reward function whose optimization would generate 'desirable' behavior."
- e.g. "Driving well"
- Apprenticeship learning: Recovering expert's underlying reward function more "parsimonious" than learning expert's policy?

Slide 7

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Applications in multi-agent systems

- In multi-agent adversarial games, learning opponents' reward functions that guide their actions to devise strategies against them.
- In mechanism design, learning each agent's reward function from histories to manipulate its actions.
- and more?

Slide 8 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

MDP Recap

- MDP is represented as a tuple $(S, A, \{P_{sa}\}, \gamma, R)$
Note: R is bounded by R_{max}
- Value function for policy π :

$$V^\pi(s_1) = E[R(s_1) + \gamma R(s_2) + \dots | \pi]$$
- Q-function:

$$Q^\pi(s, a) = R(s) + \gamma E_{s' \sim P_{sa}(\cdot)}[V^\pi(s')]$$

Slide 9 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

MDP Recap

- Bellman Equation:

$$V^\pi(s) = R(s) + \gamma \sum_{s'} P_{s\pi(s)}(s') V^\pi(s')$$

$$Q^\pi(s) = R(s) + \gamma \sum_{s'} P_{s\pi(s)}(s') V^\pi(s')$$
- In matrix notation:^{s'}
 - V is a vector of values ($|S| \times 1$)
 - R is a vector of rewards ($|S| \times 1$)
 - P is a matrix of transition probabilities ($|S| \times |S|$)
$$V^\pi = R + \gamma \cdot P_{a_1} \cdot V^\pi$$

$$V^\pi = (I - \gamma P_{a_1})^{-1} R$$

Slide 10 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

IRL: Finite State Space

- Reward function solution set (a_1 is optimal action)

$$V^\pi = R + \gamma \cdot P_{a_1} \cdot V^\pi$$

$$V^\pi = (I - \gamma P_{a_1})^{-1} R$$

$$a_1 \equiv \pi(s) \in \arg \max_{a \in A} Q^\pi(s, a)$$

$$\Leftrightarrow P_{a_1} V^\pi \geq P_a V^\pi \quad \forall a \in A \setminus a_1$$

$$\Leftrightarrow P_{a_1} (I - \gamma P_{a_1})^{-1} R \geq P_a (I - \gamma P_{a_1})^{-1} R \quad \forall a \in A \setminus a_1$$

$$\Leftrightarrow (P_{a_1} - P_a) (I - \gamma P_{a_1})^{-1} R \geq 0 \quad \forall a \in A \setminus a_1$$

Slide 11 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

IRL: Finite State Space

$$(P_{a_1} - P_a) (I - \gamma P_{a_1})^{-1} R \geq 0, \forall a \in A \setminus a_1$$

There are many solutions of R that satisfy the inequality (e.g. R = 0), which one might be the best solution?

- Make deviation from π as costly as possible:

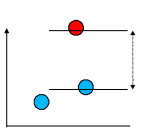
$$\sum_{s \in S} (Q^\pi(s, a_1) - \max_{a \in A \setminus a_1} Q^\pi(s, a))$$
- Make reward function as simple as possible

Slide 12 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

IRL: Finite State Space



- Linear Programming Formulation:

$$\max \sum_{i=1}^N \min_{a \in \{a_2, a_3, \dots, a_k\}} \{ (P_{a_1}(i) - P_a(i)) (I - \gamma P_{a_1})^{-1} R \} - \lambda \|R\|_1$$

$$st. (P_{a_1}(i) - P_a(i)) (I - \gamma P_{a_1})^{-1} R \geq 0 \quad \forall a \in A \setminus a_1$$

$$|R_i| \leq R_{max}, i = 1, \dots, N$$

Slide 13 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

IRL: Large State Space

- Linear approximation of reward function (in driving example, $\phi_i(s)$ basis functions can be collision, stay on right lane, ...etc)

$$R(s) = \alpha_1 \phi_1(s) + \alpha_2 \phi_2(s) + \dots + \alpha_d \phi_d(s)$$
- Let V_i^π be value function of policy π , when reward $R =$

$$V^\pi = \alpha_1 V_1^\pi + \alpha_2 V_2^\pi + \dots + \alpha_d V_d^\pi$$
- For R to make $a_1 \equiv \pi(s)$ optimal

$$E_{s' \sim p_{sa}} [V^\pi(s')] \geq E_{s' \sim p_{sa}} [V^\pi(s')] \quad \forall a \in A \setminus a_1$$

Slide 14 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

IRL: Large State Spaces

- In an infinite or large number of state space, it is usually not possible to check all constraints:

$$E_{s' \sim p_{sa}} [V^\pi(s')] \geq E_{s' \sim p_{sa}} [V^\pi(s')]$$
- Choose a finite subset S_0 from all states
- Linear Programming formulation, find α , that:

$$\max_{\alpha \in S_0} \min_{a \in \{a_2, a_3, \dots, a_k\}} \{p(E_{s' \sim p_{sa}} [V^\pi(s')] - E_{s' \sim p_{sa}} [V^\pi(s')])\}$$
- $s.t. |\alpha_i| \leq 1, i = 1, \dots, d$
 $x \geq 0, p(x) = x; \text{ otherwise } p(x) = 2x$

Slide 15 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

IRL from Sample Trajectories

- If π is only accessible through a set of sampled trajectories (e.g. driving demo)
- Assume we start from a dummy state s_0 (whose next state distribution is according to D).
- In the case that reward $R = \phi_i$ trajectory state sequence (s_0, s_1, s_2, \dots) :

$$\hat{V}_i^\pi(s_0) = \phi_i(s_0) + \gamma \phi_i(s_1) + \gamma^2 \phi_i(s_2) + \dots$$

$$\hat{V}^\pi(s_0) = \alpha_1 \hat{V}_1^\pi(s_0) + \dots + \alpha_d \hat{V}_d^\pi(s_0)$$

Slide 16 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

IRL from Sample Trajectories

- Assume we have some set of policies $\{\pi_1, \pi_2, \dots, \pi_k\}$
- Linear Programming formulation

$$\max \sum_{i=1}^k p(\hat{V}^{\pi_i}(s_0) - \hat{V}^{\pi^*}(s_0))$$
- $s.t. |\alpha_i| \leq 1, i = 1, \dots, d$
- The above optimization gives a new reward R , we then compute π_{k+1} based on R , and add it to the set of policies
- reiterate

Slide 17 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Discrete Gridworld Experiment

- 5x5 grid world
- Agent starts in bottom-left square.
- Reward of 1 in the upper-right square.
- Actions = N,W,S,E (30% chance of random)

Slide 18 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Discrete Gridworld Results

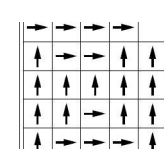
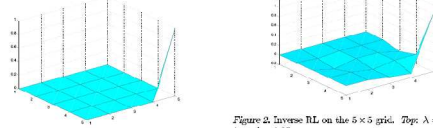



Figure 1. Top: 5x5 grid world with optimal policy. Bottom: True reward function.

Figure 2. Inverse RL on the 5x5 grid. Top: λ true $\lambda = 1.0$.

Slide 19 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Mountain Car Experiment #1

- Car starts in valley, goal is at the top of hill
- Reward is -1 per “step” until goal is reached
- State = car's x-position & velocity (continuous!)
- Function approx. class: all linear combinations of 26 evenly spaced Gaussian-shaped basis functions

Slide 20

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Mountain Car Experiment #2

- Goal is in bottom of valley
- Car starts... not sure. Top of hill?
- Reward is 1 in the goal area, 0 elsewhere
- $\gamma = 0.99$
- State = car's x-position & velocity (continuous!)
- Function approx. class: all linear combinations of 26 evenly spaced Gaussian-shaped basis functions

Slide 21

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Mountain Car Results

#1

#2

Slide 22

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Continuous Gridworld Experiment

- State space is now $[0,1] \times [0,1]$ continuous grid
- Actions: 0.2 movement in any direction + noise in x and y coordinates of $[-0.1,0.1]$
- Reward 1 in region $[0.8,1] \times [0.8,1]$, 0 elsewhere
- $\gamma = 0.9$
- Function approx. class: all linear combinations of a 15×15 array of 2-D Gaussian-shaped basis functions
- $m=5000$ trajectories of 30 steps each per policy

Slide 23

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Continuous Gridworld Results

3%-10% error when comparing fitted reward's optimal policy with the true optimal policy

However, no significant difference in quality of policy (measured using true reward function)

Figure 5. Results on the continuous grid world, for 5 runs. Top: Fraction of states on which the fitted reward's optimal policy disagrees with the true optimal policy, plotted against iteration number. Bottom: The value of the fitted reward's optimal policy. (Estimates are from 50000 Monte Carlo trials of length 50 each; negligible errorbars).

Slide 24

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Apprenticeship Learning via IRL

- For $t = 1, 2, \dots$
 - **Inverse RL step:**
 - Estimate expert's reward function $R(s) = w^T \phi(s)$ such that under $R(s)$ the expert performs better than all previously found policies $\{\pi_i\}$.
 - **RL step:**
 - Compute optimal policy π_t for
 - the estimated reward w .

Slide 25

CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Algorithm: IRL step

- Maximize $\tau, w: \|w\|_2 \leq 1 \quad \tau$
- s.t. $V_w(\pi_E) \geq V_w(\pi_i) + \tau \quad i=1, \dots, t-1$
- τ = margin of expert's performance over the performance of previously found policies.
- $V_w(\pi) = E[\sum_t \gamma^t R(s_t) | \pi] = E[\sum_t \gamma^t w^T \phi(s_t) | \pi]$
- $= w^T E[\sum_t \gamma^t \phi(s_t) | \pi]$
- $= w^T \mu(\pi)$
- $\mu(\pi) = E[\sum_t \gamma^t \phi(s_t) | \pi]$ are the "feature expectations"

Slide 26 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Feature Expectation Closeness

- If we can find a policy π such that
 - $\|\mu(\pi_E) - \mu(\pi)\|_2 \leq \epsilon,$
- then for any underlying reward $R^*(s) = w^{*T} \phi(s),$
- we have that
 - $|V_w(\pi_E) - V_w(\pi)| = |w^{*T} \mu(\pi_E) - w^{*T} \mu(\pi)|$
 - $\leq \|w^*\|_2 \|\mu(\pi_E) - \mu(\pi)\|_2$
 - $\leq \epsilon.$

Slide 27 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

IRL step as Support Vector Machine

$|w^{*T} \mu(\pi_E) - w^{*T} \mu(\pi)| = |V_{w^*}(\pi_E) - V_{w^*}(\pi)|$

Slide 28 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Gridworld Experiment

- 128 x 128 grid world divided into 64 regions, each of size 16 x 16 ("macrocells").
- A small number of macrocells have positive rewards.
- For each macrocell, there is one feature $\Phi_i(s)$ indicating whether that state s is in macrocell i
- Algorithm was also run on the subset of features $\Phi_i(s)$ that correspond to non-zero rewards.

Slide 29 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Gridworld Results

Figure 3. A comparison of the convergence speeds of the max-margin and projection versions of the algorithm on a 128x128 grid. Euclidean distance to the expert's feature expectations is plotted as a function of the number of iterations. We rescaled the feature expectations by $(1-\gamma)^t$ such that they are in $[0, 1]^k$. The plot shows averages over 40 runs, with 1 s.e. errorbars.

Distance to expert vs. # Iterations Performance vs. # Trajectories

Slide 30 CS 5300: Inverse RL

UNIVERSITY OF UTAH

Hal Daumé III (hal@cs.utah.edu)

Car Driving Experiment

- No explicit reward function at all!
- Expert demonstrates proper policy via 2 min. of driving time on simulator (1200 data points).
- 5 different "driver types" tried.
- Features: which lane the car is in, distance to closest car in current lane.
- Algorithm run for 30 iterations, policy hand-picked.
- Movie Time! (Expert left, IRL right)

Slide 31 CS 5300: Inverse RL

Car Driving Results

	Collision	Offroad Left	Left Lane	Middle Lane	Right Lane	Offroad Right
1 μ_E	0.0000	0.0000	0.1325	0.2033	0.5983	0.0658
$\mu(\bar{\pi})$	0.0001	0.0004	0.0904	0.2287	0.6041	0.0764
\bar{w}	-0.0767	-0.0439	0.0077	0.0078	0.0318	-0.0035
2 μ_E	0.1167	0.0000	0.0633	0.4667	0.4700	0.0000
$\mu(\bar{\pi})$	0.1332	0.0000	0.1045	0.3196	0.5759	0.0000
\bar{w}	0.2340	-0.1098	0.0092	0.0487	0.0576	-0.0056
3 μ_E	0.0000	0.0000	0.0000	0.0033	0.7058	0.2908
$\mu(\bar{\pi})$	0.0000	0.0000	0.0000	0.0000	0.7447	0.2554
\bar{w}	-0.1056	-0.0051	-0.0573	-0.0386	0.0929	0.0081
4 μ_E	0.0600	0.0000	0.0000	0.0033	0.2908	0.7058
$\mu(\bar{\pi})$	0.0569	0.0000	0.0000	0.0000	0.2666	0.7334
\bar{w}	0.1079	-0.0001	-0.0487	-0.0666	0.0590	0.0564
5 μ_E	0.0600	0.0000	0.0000	1.0000	0.0000	0.0000
$\mu(\bar{\pi})$	0.0542	0.0000	0.0000	1.0000	0.0000	0.0000
\bar{w}	0.0094	-0.0108	-0.2765	0.8126	-0.5099	-0.0154

Additional References and Pointers

- Policy gradient and policy search:
 - Kakade and Langford, ICML 2002
 - Bagnell, Kakade, Ng and Schneider, NIPS 2003
 - Bhatnagar, Sutton, Ghavamzadeh and Lee, NIPS 2007
 - Peters and Schaal, NN 2008
- Apprenticeship learning:
 - Ng and Russell, ICML 2000
 - Abbeel and Ng, ICML 2004
 - Ratliff, Bagnell and Zinkevich, ICML 2006
 - Daume, Langford and Marcu, MLJ 2008
 - Wingate and Singh, NIPS 2008
- Plus lots of other stuff by these guys, Rich Sutton, Andy Barto, Michael Kearns, and many others...