Inverse Reinforcement Learning

Pieter Abbeel
UC Berkeley EECS
Inverse Reinforcement Learning

[equally good titles: Inverse Optimal Control, Inverse Optimal Planning]

Pieter Abbeel
UC Berkeley EECS
High-level picture

Dynamics Model \( T \)

Reward Function \( R \)

Reinforcement Learning / Optimal Control

Controller / Policy \( \pi^* \)

Probability distribution over next states given current state and action

\[ \text{arg max}_\pi \mathbb{E}[\sum_t \gamma^t R(s_t) | \pi] \]

Inverse RL:

Given \( \pi^* \) and \( T \), can we recover \( R \)?

More generally, given execution traces, can we recover \( R \)?
Motivation for inverse RL

- Scientific inquiry
  - Model animal and human behavior
    - E.g., bee foraging, songbird vocalization. [See intro of Ng and Russell, 2000 for a brief overview.]

- Apprenticeship learning/Imitation learning through inverse RL
  - Presupposition: reward function provides the most succinct and transferable definition of the task
  - Has enabled advancing the state of the art in various robotic domains

- Modeling of other agents, both adversarial and cooperative
Lecture outline

- Example applications
- Inverse RL vs. behavioral cloning
- Historical sketch of inverse RL
- Mathematical formulations for inverse RL
- Case studies
Examples

- Simulated highway driving
  - Abbeel and Ng, ICML 2004,
  - Syed and Schapire, NIPS 2007

- Aerial imagery based navigation
  - Ratliff, Bagnell and Zinkevich, ICML 2006

- Parking lot navigation
  - Abbeel, Dolgov, Ng and Thrun, IROS 2008

- Urban navigation
  - Ziebart, Maas, Bagnell and Dey, AAAI 2008
Examples (ctd)

- Human path planning
  - Mombaur, Truong and Laumond, AURO 2009

- Human goal inference
  - Baker, Saxe and Tenenbaum, Cognition 2009

- Quadruped locomotion
  - Ratliff, Bradley, Bagnell and Chestnutt, NIPS 2007
  - Kolter, Abbeel and Ng, NIPS 2008
Urban navigation

- Reward function for urban navigation?

→ destination prediction

Ziebart, Maas, Bagnell and Dey AAAI 2008
Lecture outline

- Example applications
- *Inverse RL vs. behavioral cloning*
- Historical sketch of inverse RL
- Mathematical formulations for inverse RL
- Case studies
Problem setup

- **Input:**
  - State space, action space
  - No reward function

- **Inverse RL:**
  - Can we recover $R$?

- **Apprenticeship learning via inverse RL**
  - Can we then use this $R$ to find a good policy?

- **Behavioral cloning**
  - Can we directly learn the teacher’s policy using supervised learning?

- **Transition model** $P_{sa}(s_{t+1} \mid s_t, a_t)$

- **Teacher’s demonstration:** $s_0, a_0, s_1, a_1, s_2, a_2, ...$
  ($= \text{trace of the teacher’s policy } \pi^*$)
Behavioral cloning

- Formulate as standard machine learning problem
  - Fix a policy class
    - E.g., support vector machine, neural network, decision tree, deep belief net, ...
  - Estimate a policy (=mapping from states to actions) from the training examples \((s_0, a_0), (s_1, a_1), (s_2, a_2), \ldots\)

- Two of the most notable success stories:
  - Pomerleau, NIPS 1989: ALVINN
  - Sammut et al., ICML 1992: Learning to fly (flight sim)
Inverse RL vs. Behavioral cloning

- Which has the most succinct description: $\pi^*$ vs. $R^*$?

- Especially in planning oriented tasks, the reward function is often much more succinct than the optimal policy.
Lecture outline

- Example applications
- Inverse RL vs. behavioral cloning
- *Historical sketch of inverse RL*
- Mathematical formulations for inverse RL
- Case studies
Inverse RL history

- 1964, Kalman posed the inverse optimal control problem and solved it in the 1D input case
- 1994, Boyd+al.: a linear matrix inequality (LMI) characterization for the general linear quadratic setting
- 2000, Ng and Russell: first MDP formulation, reward function ambiguity pointed out and a few solutions suggested
- 2004, Abbeel and Ng: inverse RL for apprenticeship learning---reward feature matching
- 2006, Ratliff+al: max margin formulation
Inverse RL history

- 2007, Ratliff+al: max margin with boosting---enables large vocabulary of reward features
- 2007, Ramachandran and Amir [R&A], and Neu and Szepesvari: reward function as characterization of policy class
- 2008, Kolter, Abbeel and Ng: hierarchical max-margin
- 2008, Syed and Schapire: feature matching + game theoretic formulation
- 2008, Ziebart+al: feature matching + max entropy
- 2008, Abbeel+al: feature matching -- application to learning parking lot navigation style
- 2009, Baker, Saxe, Tenenbaum: same formulation as [R&A], investigation of understanding of human inverse planning inference
- 2009, Mombaur, Truong, Laumond: human path planning
- Active inverse RL? Inverse RL w.r.t. minmax control, partial observability, learning stage (rather than observing optimal policy), ...?
Lecture outline

- Example applications
- Inverse RL vs. behavioral cloning
- Historical sketch of inverse RL
  - Mathematical formulations for inverse RL
- Case studies
Three broad categories of formalizations

- Max margin
- Feature expectation matching
- Interpret reward function as parameterization of a policy class
Basic principle

- Find a reward function $R^*$ which explains the expert behavior.
- Find $R^*$ such that
  \[
  \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t R^*(s_t) \left| \pi^* \right. \right] \geq \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t R^*(s_t) \left| \pi \right. \right], \quad \forall \pi
  \]
- In fact a convex feasibility problem, but many challenges:
  - $R=0$ is a solution, more generally: reward function ambiguity
  - We typically only observe expert traces rather than the entire expert policy $\pi^*$ --- how to compute left-hand side?
  - Assumes the expert is indeed optimal --- otherwise infeasible
  - Computationally: assumes we can enumerate all policies
Feature based reward function

- Let $R(s) = w^\top \phi(s)$, where $w \in \mathbb{R}^n$, and $\phi : S \rightarrow \mathbb{R}^n$.

\[
E[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi] = E[\sum_{t=0}^{\infty} \gamma^t w^\top \phi(s_t)|\pi]
\]
\[
= w^\top E[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi]
\]
\[
= w^\top \mu(\pi)
\]

Expected cumulative discounted sum of feature values or “feature expectations”

- Subbing into

\[
E[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] \geq E[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi]\quad \forall \pi
\]
gives us: Find $w^*$ such that $w^*^\top \mu(\pi^*) \geq w^*^\top \mu(\pi)\quad \forall \pi$
Feature based reward function

Let \( R(s) = w^\top \phi(s) \), where \( w \in \mathbb{R}^n \), and \( \phi : S \rightarrow \mathbb{R}^n \).

Find \( w^* \) such that \( w^* \mu(\pi^*) \geq w^* \mu(\pi) \quad \forall \pi \)

- Feature expectations can be readily estimated from sample trajectories.
- The number of expert demonstrations required scales with the number of features in the reward function.
- The number of expert demonstration required does not depend on
  - Complexity of the expert’s optimal policy \( \nicefrac{\gamma^*}{4} \)
  - Size of the state space
Recap of challenges

Let \( R(s) = w^\top \phi(s) \), where \( w \in \mathbb{R}^n \), and \( \phi : S \to \mathbb{R}^n \).

\[
    w^* \mu(\pi^*) \geq w^* \mu(\pi) \quad \forall \pi
\]

**Challenges:**

- Assumes we know the entire expert policy \( \pi^* \) \( \rightarrow \) assumes we can estimate expert feature expectations
- \( R=0 \) is a solution (now: \( w=0 \)), more generally: reward function ambiguity
- Assumes the expert is indeed optimal---became even more of an issue with the more limited reward function expressiveness!
- Computationally: assumes we can enumerate all policies
Ambiguity

- **Standard max margin:**
  \[
  \min_{w} \|w\|_2^2 \\
  \text{s.t. } w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + 1 \quad \forall \pi
  \]

- **“Structured prediction” max margin:**
  \[
  \min_{w} \|w\|_2^2 \\
  \text{s.t. } w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + m(\pi^*, \pi) \quad \forall \pi
  \]

- **Justification:** margin should be larger for policies that are very different from \(\pi^*\).

- **Example:** \(m(\pi, \pi^*) = \) number of states in which \(\pi^*\) was observed and in which \(\pi\) and \(\pi^*\) disagree
Expert suboptimality

- Structured prediction max margin with slack variables:

\[
\min_{w, \xi} \|w\|_2^2 + C\xi \\
\text{s.t.} \quad w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + m(\pi^*, \pi) - \xi \quad \forall \pi
\]

- Can be generalized to multiple MDPs (could also be same MDP with different initial state)

\[
\min_{w, \xi^{(i)}} \|w\|_2^2 + C\sum_i \xi^{(i)} \\
\text{s.t.} \quad w^\top \mu(\pi^{(i)*}) \geq w^\top \mu(\pi^{(i)}) + m(\pi^{(i)*}, \pi^{(i)}) - \xi^{(i)} \quad \forall i, \pi^{(i)}
\]
Resolved: access to $\pi^*$, ambiguity, expert suboptimality

One challenge remains: very large number of constraints

- Ratliff+al use subgradient methods.
- In this lecture: constraint generation

Complete max-margin formulation

$$\min_w \|w\|^2_2 + C \sum_i \xi^{(i)}$$

s.t. $w^\top \mu(\pi^{(i)*}) \geq w^\top \mu(\pi^{(i)}) + m(\pi^{(i)*}, \pi^{(i)}) - \xi^{(i)} \quad \forall i, \pi^{(i)}$

[Ratliff, Zinkevich and Bagnell, 2006]
Constraint generation

Initialize $\Pi^{(i)} = \{}$ for all $i$ and then iterate

- Solve

$$\min_w \|w\|_2^2 + C \sum_i \xi^{(i)}$$

s.t. $w^\top \mu(\pi^{(i)*}) \geq w^\top \mu(\pi^{(i)}) + m(\pi^{(i)*}, \pi^{(i)}) - \xi^{(i)} \quad \forall i, \forall \pi^{(i)} \in \Pi^{(i)}$

- For current $w$, find most violated constraint for all $i$ by solving:

$$\max_{\pi^{(i)}} w^\top \mu(\pi^{(i)}) + m(\pi^{(i)*}, \pi^{(i)})$$

= find optimal policy for current estimate of reward (+ loss augmentation $m$)

- For all $i$ add $\pi^{(i)}$ to $\Pi^{(i)}$

- If no constraint violations were found, we are done.
Every policy $\pi$ has a corresponding feature expectation vector $\mu(\pi)$, which for visualization purposes we assume to be 2D.

$$E[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi] = w^\top \mu(\pi)$$
Every policy $\pi$ has a corresponding feature expectation vector $\mu(\pi)$, which for visualization purposes we assume to be 2D.

**Constraint generation:**

$$\max_{\pi} \mathbb{E} [\sum_{t=0}^{\infty} \gamma^t R_w(s_t)|\pi] = \max_{\pi} w^\top \mu(\pi)$$
Three broad categories of formalizations

- Max margin (Ratliff+al, 2006)
  - Feature boosting [Ratliff+al, 2007]
  - Hierarchical formulation [Kolter+al, 2008]

- Feature expectation matching (Abbeel+Ng, 2004)
  - Two player game formulation of feature matching (Syed+Schapire, 2008)
  - Max entropy formulation of feature matching (Ziebart+al, 2008)

- Interpret reward function as parameterization of a policy class. (Neu +Szepesvari, 2007; Ramachandran+Amir, 2007; Baker, Saxe, Tenenbaum, 2009; Mombaur, Truong, Laumond, 2009)
Feature matching

- Inverse RL starting point: find a reward function such that the expert outperforms other policies

\[
R(s) = w^\top \phi(s), \quad \text{where } w \in \mathbb{R}^n, \quad \text{and } \phi : S \to \mathbb{R}^n.
\]

Find \( w^* \) such that \( w^{*\top} \mu(\pi^*) \geq w^{*\top} \mu(\pi) \quad \forall \pi \)

- Observation in Abbeel and Ng, 2004: for a policy \( \pi \) to be guaranteed to perform as well as the expert policy \( \mu^* \), it suffices that the feature expectations match:

\[
\|\mu(\pi) - \mu(\pi^*)\|_1 \leq \epsilon
\]

implies that for all \( w \) with \( \|w\|_\infty \leq 1: \)

\[
|w^{*\top} \mu(\pi) - w^{*\top} \mu(\pi^*)| \leq \epsilon
\]
Apprenticeship learning [Abbeel & Ng, 2004]

- Assume $R_w(s) = w^\top \phi(s)$ for a feature map $\phi : S \rightarrow \mathbb{R}^n$.
- Initialize: pick some controller $\pi_0$.
- Iterate for $i = 1, 2, \ldots$:
  - **“Guess” the reward function:**
    Find a reward function such that the teacher maximally outperforms all previously found controllers.
    $$\max_{\gamma, w : \|w\|_2 \leq 1} \gamma$$
    $$s.t. \quad w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + \gamma \quad \forall \pi \in \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$$
  - **Find optimal control policy** $\pi_i$ for the current guess of the reward function $R_w$.
  - $\gamma \leq \varepsilon/2$ exit the algorithm.
Algorithm example run

\[
\mu \left( \pi_0 \right) \rightarrow \mu \left( \pi_1 \right) \\
\mu \left( \pi_2 \right) \rightarrow \mu \left( \pi_E \right) \rightarrow \mu \left( \pi_0 \right) \\
\mu \left( \pi_2 \right) \rightarrow \mu \left( \pi_3 \right) \\
\mu \left( \pi_0 \right) \rightarrow \mu \left( \pi_2 \right) \\
\mu \left( \pi_1 \right) \rightarrow \mu \left( \pi_E \right) \rightarrow \mu \left( \pi_1 \right)
\]
Suboptimal expert case

- Can match expert by stochastically mixing between 3 policies
- In practice: for any \( w^* \) one of \( \mu_0, \mu_1, \mu_2 \) outperforms \( \pi^* \)

\[ \rightarrow \text{pick one of them.} \]

- Generally: for k-dimensional feature space the user picks between k+1 policies
If expert suboptimal then the resulting policy is a mixture of somewhat arbitrary policies which have expert in their convex hull.

In practice: pick the best one of this set and pick the corresponding reward function.

Next:
Additional assumption: \( w \geq 0, \sum_i w_i = 1. \)

Any policy in this area performs at least as well as expert.

How to find policy on pareto optimal curve in this area + corresponding reward function?

\[
E[\sum_{t=0}^\infty \gamma^t R(s_t)|\pi] = w^\top \mu(\pi)
\]
Min-Max feature expectation matching [Syed and Schapire (2008)]

Additional assumption: $w \geq 0, \sum_i w_i = 1.$
Min max games

- Example of standard min-max game setting:

  rock-paper-scissors pay-off matrix:

  \[
  \begin{array}{ccc}
  \text{maximizer} & \text{rock} & \text{paper} & \text{scissors} \\
  \text{rock} & 0 & 1 & -1 \\
  \text{paper} & -1 & 0 & 1 \\
  \text{scissors} & 1 & -1 & 0 \\
  \end{array}
  \]

  \[
  \min_{w_m : w_m \geq 0, \|w_m\|_1 = 1} \max_{w_M : w_M \geq 0, \|w_M\|_1 = 1} w_m^T G w_M
  \]
Min-Max feature expectation matching [Syed and Schapire (2008)]

- **Standard min-max game:**
  \[
  \min_{w_m : w_m \geq 0, \|w_m\|_1=1} \max_{w_M : w_M \geq 0, \|w_M\|_1=1} w_m^\top G w_M
  \]

- **Min-max inverse RL:**
  \[
  \min_{w : \|w\|_1=1, w \geq 0} \max_{\pi} w^\top (\mu(\pi) - \mu(\pi^*))
  \]

- **Solution:** maximize over weights \(w\) which weigh the contribution of all policies \(\mu_1, \mu_2, ..., \mu_N\) to the mixed policy.

- **Formally:**
  \[
  \min_w \max_{\lambda} w^\top G \lambda \quad G_{ij} = (\mu(\pi_j) - \mu(\pi^*))_i
  \]
Recall feature matching in suboptimal expert case:

$\mu(\pi_0)$, $\mu(\pi_2)$, $\mu(\pi_3)$, $\mu(\pi_4)$, $\mu(\pi_5)$, $\mu(\pi^*)$
Maximum-entropy feature expectation matching --- [Ziebart+al, 2008]

- Maximize entropy of distributions over paths followed while satisfying the constraint of feature expectation matching:

\[
\max_P \quad - \sum_{\zeta} P(\zeta) \log P(\zeta)
\]

\[
\text{s.t.} \quad \sum_{\zeta} P(\zeta) \mu(\zeta) = \mu(\pi^*)
\]

- This turns out to imply that \( P \) is of the form:

\[
P(\zeta) = \frac{1}{Z(w)} \exp(w^T \mu(\zeta))
\]
Feature expectation matching

- If expert suboptimal:
  - *Abbeel and Ng, 2004*: resulting policy is a mixture of policies which have expert in their convex hull---In practice: pick the best one of this set and pick the corresponding reward function.
  - *Syed and Schapire, 2008* recast the same problem in game theoretic form which, at cost of adding in some prior knowledge, results in having a unique solution for policy and reward function.
  - *Ziebart+al, 2008* assume the expert stochastically chooses between paths where each path’s log probability is given by its expected sum of rewards.
Lecture outline

- Example applications
- Inverse RL vs. behavioral cloning
- Historical sketch of inverse RL
- Mathematical formulations for inverse RL
  - Max-margin
  - Feature matching
  - *Reward function parameterizing the policy class*
- Case studies
Recall:

\[ V^*(s; R) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a)V^*(s; R) \]

\[ Q^*(s, a; R) = R(s) + \gamma \sum_{s'} P(s'|s, a)V^*(s; R) \]

Let’s assume our expert acts according to:

\[ \pi(a|s; R, \alpha) = \frac{1}{Z(s; R, \alpha)} \exp(\alpha Q^*(s, a; R)) \]

Then for any \( R \) and \( \alpha \), we can evaluate the likelihood of seeing a set of state-action pairs as follows:

\[ P((s_1, a_1)) \ldots P((s_m, a_m)) = \frac{1}{Z(s_1; R, \alpha)} \exp(\alpha Q^*(s_1, a_1; R)) \ldots \frac{1}{Z(s_m; R, \alpha)} \exp(\alpha Q^*(s_m, a_m; R)) \]
Reward function parameterizing the policy class

- Assume our expert acts according to:

  \[ \pi(a|s; R, \alpha) = \frac{1}{Z(s; R, \alpha)} \exp(\alpha Q^*(s, a; R)) \]

- Then for any \( R \) and \( \alpha \), can evaluate likelihood of set of state-action pairs:

  \[
P((s_1, a_1)) \cdots P((s_m, a_m)) = \frac{1}{Z(s_1; R, \alpha)} \exp(\alpha Q^*(s_1, a_1; R)) \cdots \frac{1}{Z(s_m; R, \alpha)} \exp(\alpha Q^*(s_m, a_m; R))
  \]

- Ramachandran and Amir, AAAI2007: MCMC method to sample from this distribution

- Neu and Szepesvari, UAI2007: gradient method to optimize likelihood [MAP]

- Baker, Saxe and Tenenbaum, Cognition 2009: only 3 possible reward functions \( \rightarrow \) tractable exact Bayesian inference
Reward function parameterizing the policy class -- deterministic systems

- Assume deterministic system $x_{t+1} = f(x_t, u_t)$ and an observed trajectory $(x_0^*, x_1^*, ..., x_T^*)$

- Find reward function by solving:

$$
\min_w \sum_{t=0}^{T} \|x_t^* - x_t^w\|_2 \\
\text{s.t. } x^w \text{ is the solution of:}
$$

$$
\max_x \sum_{t=0}^{T} \sum_i w_i \phi_i(x_t) \\
\text{s.t. } x_{t+1} = f(x_t, u_t) \\
x_0 = x_0^*, \quad x_T = x_T^*
$$

[Mombaur, Truong, Laumond, 2009]
Lecture outline

- Example applications
- Inverse RL vs. behavioral cloning
- History of inverse RL
- Mathematical formulations for inverse RL

- Case studies: (1) Highway driving, (2) Crusher, (3) Parking lot navigation, (4) Route inference, (5) Human path planning, (6) Human inverse planning, (7) Quadruped locomotion
Simulated highway driving

Abbeel and Ng, ICML 2004; Syed and Schapire, NIPS 2007
Highway driving

Teacher in Training World

Learned Policy in Testing World

- **Input:**
  - Dynamics model / Simulator \( P_{sa}(s_{t+1} | s_t, a_t) \)
  - Teacher’s demonstration: 1 minute in “training world”
  - Note: \( R^* \) is unknown.
  - Reward features: 5 features corresponding to lanes/shoulders; 10 features corresponding to presence of other car in current lane at different distances

[Abbeel and Ng 2004]
More driving examples

In each video, the left sub-panel shows a demonstration of a different driving “style”, and the right sub-panel shows the behavior learned from watching the demonstration.

[Abbeel and Ng 2004]
Crusher
Learning movie

data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAfQAAAA3CAIAAABxJzBjAAAACXBIWXMAAAsTAAALEwEAmpwYAAAD2UlEQVR423aQjDgDgEYdJpZB4n2oAAAAASUVORK5CYII=
Max margin

[Ratliff + al, 2006/7/8]
Parking lot navigation

- Reward function trades off:
  - Staying “on-road,”
  - Forward vs. reverse driving,
  - Amount of switching between forward and reverse,
  - Lane keeping,
  - On-road vs. off-road,
  - Curvature of paths.

[Abbeel et al., IROS 08]
Experimental setup

- Demonstrate parking lot navigation on “train parking lots.”

- Run our apprenticeship learning algorithm to find reward function.

- Receive “test parking lot” map + starting point and destination.

- Find the trajectory that maximizes the learned reward function for navigating the test parking lot.
Nice driving style
Sloppy driving-style
“Don’t mind reverse” driving-style
Only 35% of routes are “fastest”
(Letchner, Krumm, & Horvitz 2006)
Time

Money

Stress

Skill

Ziebart+al, 2007/8/9
Distance, Speed, Type, Lanes, Turns, Context, Time, Fuel, Safety, Stress, Skill, Mood

Ziebart+al, 2007/8/9
Data Collection

Length, Speed, Road Type, Lanes, Accidents, Construction, Congestion, Time of day

25 Taxi Drivers

Over 100,000 miles

Ziebart+al, 2007/8/9
Destination Prediction
Human path planning

- Reward features:
  - Time to destination
  - (Forward acceleration)$^2$
  - (Sideways acceleration)$^2$
  - (Rotational acceleration)$^2$
  - Integral (angular error)$^2$

[Mombaur, Truong, Laumond, 2009]
Human path planning

- Result:
  - Time to destination:
    - $(\text{Forward acceleration})^2$ 1.2
    - $(\text{Sideways acceleration})^2$ 1.7
    - $(\text{Rotational acceleration})^2$ 0.7
    - Integral (angular error)$^2$ 5.2

[Mombaur, Truong, Laumond, 2009]
Human path planning

[Mombaur, Truong, Laumond, 2009]
Human path planning

[Mombaur, Truong, Laumond, 2009]
Goal inference

- Observe partial paths, predict goal. Goal could be A, B, or C.
- + HMM-like extension: goal can change (with some probability over time).

[Baker, Saxe, Tenenbaum, 2009]
Goal inference

- Observe partial paths, predict goal. Goal could be A, B, or C.
Quadruped

- Reward function trades off 25 features.

Hierarchical max margin [Kolter, Abbeel & Ng, 2008]
Experimental setup

- Demonstrate path across the “training terrain”

- Run our apprenticeship learning algorithm to find the reward function

- Receive “testing terrain”---height map.

- Find the optimal policy with respect to the *learned reward function* for crossing the testing terrain.
Without learning
With learned reward function
Quadruped: Ratliff + al, 2007

- Run footstep planner as expert (slow!)
- Run boosted max margin to find a reward function that explains the center of gravity path of the robot (smaller state space)
- At control time: use the learned reward function as a heuristic for A* search when performing footstep-level planning
Summary

- Example applications
- Inverse RL vs. behavioral cloning
- Sketch of history of inverse RL
- Mathematical formulations for inverse RL
- Case studies

- Open directions: Active inverse RL, Inverse RL w.r.t. minmax control, partial observability, learning stage (rather than observing optimal policy), ... ?