Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Distance Minimization for Reward Learning from Scored Trajectories

Benjamin Burchfiel, Carlo Tomasi, and Ronald Parr
Duke University, Department of Computer Science
308 Research Drive
Durham, NC USA

Abstract

Many planning methods rely on the use of an immediate reward function as a portable and succinct representation of desired behavior. Rewards are often inferred from demonstrated behavior that is assumed to be near-optimal. We examine a framework, Distance Minimization IRL (DM-IRL), for learning reward functions from scores an expert assigns to possibly suboptimal demonstrations. By changing the expert's role from a demonstrator to a judge, DM-IRL relaxes some of the assumptions present in IRL, enabling learning from the scoring of arbitrary demonstration trajectories with unknown transition functions. DM-IRL complements existing IRL approaches by addressing different assumptions about the expert. We show that DM-IRL is robust to expert scoring error and prove that finding a policy that produces maximally informative trajectories for an expert to score is strongly NP-hard. Experimentally, we demonstrate that the reward function DM-IRL learns from an MDP with an unknown transition model can transfer to an agent with known characteristics in a novel environment, and we achieve successful learning with limited available training data.

Introduction

Real-world planning tasks in unstructured domains require agents that are adaptable to their settings. Hand-designing reward functions to induce desired behavior is often extremely difficult; improperly specified reward functions can lead to unexpectedly poor results. Furthermore, in situations where there is no naturally occurring immediate reward feedback, applying traditional Reinforcement Learning (RL) algorithms is not feasible. In Learning from Demonstration (LfD), the agent learns from an expert who repeatedly demonstrates how to perform a task well. This is a natural way to transfer knowledge to the learning agent without programming task behavior explicitly. Furthermore, in these situations, LfD provides a compelling approach to specifying the agent's behavior.

LfD is commonly implemented using Markov Decision Processes (MDPs), in which an agent follows a behavior policy resulting in state-space trajectories that accrue maximal reward. One option is to learn the expert's policy directly, allowing the learner to imitate the expert under similar circumstances. An alternative option, termed Inverse Reinforcement Learning (IRL), learns the expert's reward function and then computes a policy that maximizes accrued reward. This approach often offers advantages in portability in that the reward function can transfer to parts of the state space with different dynamics and different optimal actions.

We examine a learning method, Distance Minimization (DM-IRL) (El Asri, Laroche, and Pietquin 2013), that estimates the expert's reward function from single scores an expert assigns demonstrated trajectories upon their completion. Demonstration scoring has a strong precedent in human learning and performance evaluation. It is common for experiences to be judged on a 1-10 scale, for athletic performances to be scored, and for scores to be assigned to demonstrations based on empirical outcomes. In all these cases, the judge need not be good at the desired task or familiar with its intricacies at each step; The judge must simply be able to assign a reliable numerical score to demonstration trajectories performed by herself, the learner, or others. This also avoids estimating the transition function of a (human) expert, requiring only the robot's transition function be known. DM-IRL models the expert's hidden reward function as a linear combination of state features, and estimates its coefficients by (regularized) linear regression.

We provide a theoretical analysis of DM-IRL's performance when scoring noise is introduced, a proof of the strong NP-hardness of generating policies with a specific target feature expectation, an empirical examination of DM-IRL's behavior in a terrain navigation setting under a variety of conditions, and an analysis of the relation of DM-IRL to previous IRL approaches. We show that DM-IRL is robust to suboptimal demonstration and noisy scoring, even with few demonstrations, and that the rewards DM-IRL learns from MDPs with unknown dynamics can be transferred to other MDPs.

Related Work

Many influential approaches to IRL (Ng and Russell 2000; Abbeel and Ng 2004; Ramachandran and Amir 2007; Kolter, Abbeel, and Ng 2007; Syed and Schapire 2007; Syed, Bowling, and Schapire 2008; Ziebart et al. 2008; Boularias et al. 2011) assume that the reward function is lin-

ear in a set of features assigned to states and seek to match the expert's policy in feature expectation. Since rewards are a function of features, matching feature expectations implies matching expert performance. Other approaches learn a mapping from features to reward function directly—without going through feature expectations—by minimizing some measure of loss (Ratliff, Bagnell, and Zinkevich 2006; Ratliff et al. 2007; Levine, Popovic, and Koltun 2011). Either way, although some formulations of IRL model noise in expert demonstrations (Ratliff, Bagnell, and Zinkevich 2006), suboptimal demonstrations may lead to suboptimal behavior, potentially dramatically suboptimal behavior depending upon the IRL method being employed.

In active IRL, the learner queries the expert for the optimal action to take from a given state (Lopes, Melo, and Montesano 2009; Akrour et al. 2013), thereby reducing the number of demonstrations required for learning. Closely related to DM-IRL in its setting, Daniel *et al.* (Daniel et al. 2014) propose an active learning method that produces trajectories for an expert to score and learns a function that predicts the score for a whole trajectory. They then search for a policy directly, without computing the reward function. In contrast, and like conventional IRL, DM-IRL (El Asri, Laroche, and Pietquin 2013) learns an immediate state reward function.

A fundamental difference exists between DM-IRL and traditional IRL approaches. While most IRL methods attempt to learn a reward function that explains observed behavior, DM-IRL directly attempts to regress the expert's actual reward function. A learned reward function may perfectly explain observed behavior, but still fail to generalize; this is not an issue if the expert's actual reward function is learned. Table 1 provides more comparisons of DM-IRL and traditional IRL.

Method

A discrete, finite MDP is a tuple (S, A, T, D, γ, R) where S is a finite set of states and A is a finite set of actions. $T(s, a)$ is the state transition function which, given a current state and an action, returns a probability distribution over possible next states. D is a probability distribution over possible starting states, s_0, which we assume to be fixed, and $\gamma \in [0, 1)$ is the discount factor which causes future rewards to be weighted less. $R(s) \mapsto \mathbb{R}$ is the immediate reward for being in state s.

The feature function $\boldsymbol{\phi} : S \to \mathbb{R}^k$ maps a state to a k-dimensional feature vector. We assume that there exist unknown, true reward weights $\breve{\mathbf{w}} \in \mathbb{R}^k$ and a true reward function linear in the features, $\breve{R}(s) = \breve{\mathbf{w}}^T \boldsymbol{\phi}(s)$.

A policy π maps states to distributions over actions. Its expected value with respect to $\breve{\mathbf{w}}$ is

$$V_\pi = E\left[\sum_{t=0}^{\infty} \gamma^t \breve{R}(s_t) \mid \pi\right] = \breve{\mathbf{w}}^T \sum_{t=0}^{\infty} \gamma^t E[\boldsymbol{\phi}(s_t) \mid \pi] . \quad (1)$$

We can simplify this expression further by defining the *discounted feature expectations*

$$\boldsymbol{\mu}(\pi) = \sum_{t=0}^{\infty} \gamma^t E[\boldsymbol{\phi}(s_t) \mid \pi] \in \mathbb{R}^k \quad (2)$$
$$\text{so that} \quad V_\pi = \breve{\mathbf{w}}^T \boldsymbol{\mu}(\pi) .$$

The optimal policy with respect to reward vector $\mathbf{w}$ is

$$\pi^*_{\mathbf{w}} = \arg\max_{\pi} \ \mathbf{w}^T \boldsymbol{\mu}(\pi).$$

Distance Minimization Inverse RL (DM-IRL)

Rather than requiring an expert to perform a desired task optimally, DM-IRL assumes that the expert is capable of assigning numerical scores to demonstrations. These trajectory returns, similar in concept to scores a judge might assign an Olympic performance or a food critic might assign to a dish, constitute the signal from which DM-IRL infers the reward function. We also assume that the learner has access to the *trajectories* for each demonstration—sequences of state-action pairs.

The goal is to infer the weights $\hat{\mathbf{w}}$ of a plausible reward function $\hat{R}(s) = \hat{\mathbf{w}}^T \boldsymbol{\phi}(s)$ to explain the expert's scores. Let

$$\boldsymbol{\psi}(\tau_i) = \sum_{t=0}^{|\tau_i|-1} \gamma^t \boldsymbol{\phi}(s_t(\tau_i))$$

be the *discounted accrued features* for trajectory τ_i where $s_t(\tau_i)$ denotes the state occurring at time t in τ_i. We assume the true trajectory scores to be

$$\begin{aligned} v_i &= \sum_{t=0}^{|\tau_i|-1} \gamma^t \breve{\mathbf{w}}^T \boldsymbol{\phi}(s_t(\tau_i)) \\ &= \breve{\mathbf{w}}^T \sum_{t=0}^{|\tau_i|-1} \gamma^t \boldsymbol{\phi}(s_t(\tau_i)) \\ &= \breve{\mathbf{w}}^T \boldsymbol{\psi}(\tau_i). \end{aligned}$$

Since the same reward weight vector appears in the immediate reward function and in the trajectory score function, we can determine the former if we can estimate the weights for the latter.

Scores produced by human judges may be noisy for reasons including intrinsic inconsistency, judging bias, or score quantization. Such quantization may derive from scoring rules imposed on the judges, such as food ratings of one to five stars. This rounding introduces additive error bounded by half the width of the quantization bins. We collect the scores $\widetilde{v}_i = v_i + \eta_i$ from the expert—marred by noise η_i from the aforementioned causes—into a vector $\widetilde{\mathbf{v}}$. For the remainder of this paper, we denote true (non-noisy) scores $\breve{\mathbf{v}}$.

We note that DM-IRL requires the score provided by the expert to approximate the sum of the expert's (hidden) immediate rewards. This is the natural counterpart to the assumption made in RL and IRL, that the quantity we wish to maximize is the (discounted) sum of rewards in an MDP. If

traditional IRL is applied to an expert who is not maximizing their internal sum of rewards, it will fail to learn an appropriate policy. This is easy to see as solving an MDP for π^* explicitly finds the policy that maximizes the expected discounted sum of rewards. Similar to existing RL and IRL methods that make use of MDPs, DM-IRL assumes that the performance criterion for the MDP depends upon the discounted sum of rewards. Unlike most other methods, however, it does not make any assumptions about whether the reward should be maximized or minimized when learning the reward.

Given r input trajectories and k features, define an $r \times k$ matrix M whose rows $\mathbf{m}_i^T = \boldsymbol{\psi}(\tau_i)^T$ hold discounted trajectory feature sums. We compute the estimate $\hat{\mathbf{w}}$ by linear regression:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \ ||M\mathbf{w} - \widetilde{\mathbf{v}}||,$$

where $\| \cdot \|$ could be the 2-norm for noisy problems, or ∞-norm for deterministic domains.

Learning from Noisy Scores

We now examine the effects of scoring noise on the learned solution. Because DM-IRL does not make any assumptions on the distribution from which demonstration trajectories are drawn, finite sample analysis is not practical. Since the expert may not be performing the trajectories herself and may not have any control over the policy executed by the demonstrator, there are not many plausible assumptions that could be made that would lead to a satisfying finite sample analysis. Instead, we bound the suboptimality (with respect to the true value function) of the policy that is optimal with respect to the learned reward function. Three things contribute to this bound: 1) the maximum norm of the *empirical* regression error, 2) the maximum noise in the reviewer score, and 3) the proximity of the optimal policy's feature distribution to the convex hull of the training data. This provides a useful *a posteriori* bound since all but the expert scoring noise are easily calculated after the algorithm is run. We could use L_∞ regression to couple the regression perfectly with the analysis. In our experiments, however, we take the typical approach of using L_2 regression for its computational efficiency and greater robustness to outliers than L_∞.

The *true evaluation function* $\breve{e}(\mathbf{p})$ maps a point, $\mathbf{p}$, in k dimensional feature space to a score corresponding to the desirability of $\mathbf{p}$,

$$\breve{e}(\mathbf{p}) = \breve{\mathbf{w}}^T\mathbf{p}\ , \tag{3}$$

and the *learned evaluation function* is

$$\hat{e}(\mathbf{p}) = \hat{\mathbf{w}}^T\mathbf{p}. \tag{4}$$

We use (3) to rewrite the state reward function, trajectory score function, and value function:

$$\breve{R}(s) = \breve{\mathbf{w}}^T\boldsymbol{\phi}(s) = \breve{e}(\boldsymbol{\phi}(s)) \tag{5}$$

$$v(\tau) = \breve{\mathbf{w}}^T\boldsymbol{\psi}(\tau) = \breve{e}(\boldsymbol{\psi}(\tau)) \tag{6}$$

$$V_\pi(D) = \breve{\mathbf{w}}^T\boldsymbol{\mu}(\pi) = \breve{e}(\boldsymbol{\mu}(\pi)). \tag{7}$$

Let $\delta = ||\breve{\mathbf{v}} - \widetilde{\mathbf{v}}||_\infty$ be the largest magnitude scoring noise introduced by the expert and $\epsilon = ||M\hat{\mathbf{w}} - \widetilde{\mathbf{v}}||_\infty$ be the L_∞ regression error. Any point $\mathbf{p}$ in the row space of M may be written as a linear combination of the rows of M, and if $\text{rank}\ (M) > r$ there are infinitely many choices of combination coefficients for any such $\mathbf{p}$. One choice that facilitates the derivation of bounds is a solution to the following LP:[1]

$$\min_{\alpha_1,\ldots,\alpha_r} \ \sum_{i=1}^{r} \alpha_i \tag{8}$$

such that

$$\mathbf{p} = \sum_{i=1}^{r} \alpha_i \mathbf{m_i}^T \quad \text{and} \quad \alpha_i \geq 0 \quad \forall i \in \{1, ..., r\}\ .$$

Theorem 1. *Let $\mathbf{p}$ be a point in the row space of M. Let $\alpha = \sum_{i=1}^{r} \alpha_i$ as determined by solving problem (8). The learner's evaluation error of $\mathbf{p}$ is bounded as* [2]

$$|\breve{e}(\mathbf{p}) - \hat{e}(\mathbf{p})| \leq \alpha(\delta + \epsilon).$$

Proof: The true score for $\mathbf{p}$ is

$$\begin{aligned}\breve{e}(\mathbf{p}) = \breve{\mathbf{w}}^T\mathbf{p} &= \breve{\mathbf{w}}^T \sum_{i=1}^{r} \alpha_i \mathbf{m_i}^T \\ &= \sum_{i=1}^{r} \alpha_i \breve{\mathbf{w}}^T \mathbf{m_i}^T = \sum_{i=1}^{r} \alpha_i \breve{v}_i.\end{aligned}$$

By definition, the learner's score for point $\mathbf{p}$ is

$$\hat{e}(\mathbf{p}) = \hat{\mathbf{w}}^T\mathbf{p} = \hat{\mathbf{w}}^T \sum_{i=1}^{r} \alpha_i \mathbf{m_i}^T.$$

Furthermore, by the definition of regression error, ϵ

$$\widetilde{v}_i - \epsilon \leq \hat{\mathbf{w}}^T \mathbf{m_i}^T \leq \widetilde{v}_i + \epsilon \ \ \forall i \in \{1, ..., r\}$$

and thus

$$\sum_{i=1}^{r} \alpha_i(\widetilde{\mathbf{v}} - \epsilon) \leq \hat{\mathbf{w}}^T \sum_{i=1}^{r} \alpha_i \mathbf{m_i}^T \leq \sum_{i=1}^{r} \alpha_i(\widetilde{\mathbf{v}} + \epsilon).$$

We want to recover a bound on the error $|\breve{e}(\mathbf{p}) - \hat{e}(\mathbf{p})|$ so

$$\begin{aligned}|\breve{e}(\mathbf{p}) - \hat{e}(\mathbf{p})| \ &= \ \left| \sum_{i=1}^{r} \alpha_i \breve{v}_i - \hat{\mathbf{w}}^T \sum_{i=1}^{r} \alpha_i \mathbf{m_i}^T \right| \\ &\leq \ \left| \sum_{i=1}^{r} \alpha_i \breve{v}_i - \sum_{i=1}^{r} \alpha_i \widetilde{v}_i \right| + \sum_{i=1}^{r} \alpha_i \epsilon \\ &\leq \ \sum_{i=1}^{r} \alpha_i \delta + \sum_{i=1}^{r} \alpha_i \epsilon = \sum_{i=1}^{r} \alpha_i(\delta + \epsilon) \\ &= \ \alpha(\delta + \epsilon). \quad \square\end{aligned}$$

[1] The constraint on the sign of each α_i is without loss of generality if for every discounted trajectory feature sum and score we also add the negation of both feature sum and score to the training set. In practice, features are often constrained to have only positive values in which case this is not required.

[2] L_∞ regression can also be employed to explicitly minimize ϵ when solving for $\hat{\mathbf{w}}$.

Table 1: Conceptual Comparison of Approaches

	DM-IRL	TRADITIONAL IRL
EXPERT REQUIREMENTS	SCORES TRAJECTORIES	PERFORMS TRAJECTORIES
OPTIMALITY ASSUMPTION	NEAR-OPTIMAL SCORER	NEAR-OPTIMAL DEMONSTRATOR
MDP REQUIREMENTS	SOLVE ONCE AFTER $R(s)$ IS FOUND	SOLVE TO FIND $R(s)$, THEN SOLVE TO FIND π^*
TRANSITION FUNCTIONS NEEDED	LEARNER'S	LEARNER'S AND EXPERT'S
LEARNED REWARD FUNCTION	THE EXPERT'S	INDUCES EXPERT'S BEHAVIOR

Theorem 2. *Let $\pi^*_{\breve{\mathbf{w}}}$ denote the optimal policy with respect to the true reward vector $\breve{\mathbf{w}}$ and let $\pi^*_{\hat{\mathbf{w}}}$ denote the optimal policy with respect to the learned reward vector $\hat{\mathbf{w}}$. If $\boldsymbol{\mu}(\pi^*_{\breve{\mathbf{w}}})$ and $\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}})$ both lie inside the convex hull of the rows of M, then*

$$V_{\pi^*_{\breve{\mathbf{w}}}} - V_{\pi^*_{\hat{\mathbf{w}}}} \leq 2(\delta + \epsilon).$$

Proof: We apply Theorem 1, noting that $\alpha \leq 1$ due to the assumption that $\pi^*_{\breve{\mathbf{w}}}$and $\pi^*_{\hat{\mathbf{w}}}$lie in the convex hull of the rows of M. Thus,

$$|\breve{e}(\boldsymbol{\mu}(\pi^*_{\breve{\mathbf{w}}})) - \hat{e}(\boldsymbol{\mu}(\pi^*_{\breve{\mathbf{w}}}))| \leq \delta + \epsilon$$
and
$$|\breve{e}(\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}})) - \hat{e}(\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}}))| \leq \delta + \epsilon.$$

Due to the optimality of $\pi^*_{\hat{\mathbf{w}}}$with respect to $\hat{\mathbf{w}}$,

$$\hat{e}(\boldsymbol{\mu}(\pi^*_{\breve{\mathbf{w}}})) \leq \hat{e}(\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}})).$$

And so, from equation (7),

$$\begin{aligned} V_{\pi^*_{\breve{\mathbf{w}}}} - V_{\pi^*_{\hat{\mathbf{w}}}} &= \breve{e}(\boldsymbol{\mu}(\pi^*_{\breve{\mathbf{w}}})) - \breve{e}(\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}})) \\ &\leq [\hat{e}(\boldsymbol{\mu}(\pi^*_{\breve{\mathbf{w}}})) + (\delta+\epsilon)] - [\hat{e}(\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}})) - (\delta+\epsilon)] \\ &\leq [\hat{e}(\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}})) + (\delta+\epsilon)] - [\hat{e}(\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}})) - (\delta+\epsilon)] \\ &= 2(\delta+\epsilon). \quad \square \end{aligned}$$

Learning from Limited Trajectories

With k features, DM-IRL requires at least k linearly independent trajectories to ensure rank $M \geq k$ and a unique solution. In complex domains, k may be large, and it may not be clear *a priori* exactly which subset of all features is informative. It may thus be desirable to include a large pool of features but still require few trajectories to be scored. In our experiments, we use LASSO (Tibshirani 1996),

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \ ||M\mathbf{w} - \widetilde{\mathbf{v}}||_2 + \lambda||\mathbf{w}||_1 \ ,$$

to infer $\hat{\mathbf{w}}$ when rank $M \leq k$. By using L_1 regularization as a computationally efficient proxy for L_0 regularization, LASSO tends to sharply reduce the number of non-zero elements in the solution vector. While in general the parameter λ may have to be set by cross-validation, we obtained good results in our experiments by manually setting λ to a small value[3] relative to the expert scores.

[3]In our experiments, $\lambda = 10^{-3}$.

Hardness of Targeted Policy Feature Expectations

Because the demonstrator and scorer are decoupled, DM-IRL allows the learner to create trajectories for the expert to score. This ability seems particularly compelling in trajectory-limited situations in which the demonstrator may wish to create trajectories with feature expectations unlike those previously seen. Unfortunately, it has been previously shown that finding a policy such that $\boldsymbol{\mu}(\pi) = \mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^k$ is a specific feature target, is weakly NP-Complete by reduction from the subset-sum problem (Chatterjee, Majumdar, and Henzinger 2006). We strengthen this result and prove that

Theorem 3. *Finding a policy such that $\boldsymbol{\mu}(\pi) = \mathbf{x}$ is strongly NP-Hard.*

Proof: We show that 3-SAT reduces to finding $\boldsymbol{\mu}(\pi) = \mathbf{x}$ on some MDP. Given an instance of 3-SAT (Karp 1972) with n clauses and m variables, create an MDP as follows:
1) Define m features with feature i corresponding to the ith variable in the 3-SAT instance.
2) Create a starting state, s_0, with all 0 features.
3) For each clause c_i, create state s_{c_i}, all with all 0 features and $P(s_{c_i}|s_0) = \frac{1}{n}$.
4) Create 3 states for each clause, 1 for each variable occurring in that clause, $s^{vj}_{ci}, s^{vk}_{ci}, s^{vl}_{ci}$. For each such state, if the variable is negated in that particular clause, set the corresponding feature for that variable equal to 1, otherwise set the corresponding feature to -1. Set all other features to 0. Each s_{ci} has three actions defined that deterministically transition to one of $\{s^{vj}_{ci}, s^{vk}_{ci}, s^{vl}_{ci}\}$.
5) Create a state, s_{vh}, for each variable $h \in \{1, ..., m\}$. Each such state has all 0 features. Each s^{vh}_{ci} $\forall i \in \{1, ..., n\}$ $\forall h \in \{1, ..., m\}$ has a single deterministic transition to s_{vh}.
6) Create 2 more states, s^+_{vh} and s^-_{vh}, for each variable $h \in \{1, ..., m\}$. For each positive state, set the feature corresponding to variable h to 1 and for each such negative state set the corresponding feature to -1. Each s_{vh} has 2 actions which deterministically transition to s^+_{vh} and s^-_{vh}.
7) Create a single absorbing state, s_E, with all 0 features that transitions to itself with probability 1 regardless of the action taken.
8) Let $\gamma = 1$.

Given such a construction, finding a solution to the resulting MDP such that $\boldsymbol{\mu}(\pi, s_0) = \bar{\mathbf{0}}$ yields a solution to the 3-SAT instance. Furthermore, any solution to the 3-SAT instance yields a policy π such that $\boldsymbol{\mu}(\pi, s_0) = \bar{\mathbf{0}}$. Thus, 3-SAT reduces to finding a policy such that $\boldsymbol{\mu}(\pi, s_0) = \bar{\mathbf{0}}$.

The reduction also works for other target values of $\boldsymbol{\mu}(\pi)$ and/or $\gamma < 1$ by adjusting the feature values for the states. See Figure 2(d) for the MDP construction for a simple 2 clause instance.

Experimental Results

Inferring a reward function from demonstration is often an ill-posed problem; there can be many (even infinitely many) reward functions consistent with a given set of demonstrations (Ng, Harada, and Russell 1999). Comparing learned reward weights directly is often uninformative as very different weight vectors may induce the same policy. Instead, we define a scoring function that evaluates estimated rewards through the values of their induced optimal policies:[4]

$$f(\hat{\mathbf{w}}) = \frac{\breve{e}(\boldsymbol{\mu}(\pi^*_{\hat{\mathbf{w}}}))}{\breve{e}(\boldsymbol{\mu}(\pi^*_{\breve{\mathbf{w}}}))}.$$

This scoring function yields values from 0 to 1 inclusive, with 1 indicating performance equivalent to that of the true reward weights $\breve{\mathbf{w}}$. In our evaluations, we set the initial state distribution D to the uniform distribution over all states.

A direct comparison of DM-IRL to existing methods is problematic. DM-IRL requires judging trajectories, not performing them, it is relatively agnostic to the distribution from which trajectories are drawn, makes no assumption of knowledge about the demonstrator's transition function or initial state distribution, and approximates the expert's actual reward function. Because DM-IRL is applicable to a unique set of circumstances, an apples to apples comparison with existing work is not possible. Instead, our experimental section is intended to serve as a proof of concept for DM-IRL. We apply DM-IRL to a variety of circumstances to characterize its behavior when available trajectories are limited, the number of features becomes large, and expert scoring noise grows.

Satellite Imagery Terrain Navigation

To evaluate DM-IRL in a real-world domain, we learn terrain navigation in satellite imagery. We discretize an overhead image into a 45×30 grid of 25-by-25-pixel squares, each square corresponding to a state in the MDP. The resulting 1350 states were manually labeled with rewards to provide evaluation ground truth and as the basis for a synthetic expert.[5] As in Daniel *et al.* (Daniel et al. 2014), we produce trajectory scores by computing $\breve{\mathbf{v}}$ and then adding noise to obtain $\widetilde{\mathbf{v}}$. The use of a synthetic expert allows for careful control of the amounts of noise present in the scoring, making it clear how DM-IRL responds as noise increases allowing for better characterization of DM-IRL's performance.

[4]This scoring function assumes that the evaluation function $e(x)$ returns non-negative values. To ensure this, we compute $\breve{R}(s) = \breve{\mathbf{w}}^T \boldsymbol{\phi}(s)\ \forall s \in S$ and then subtract the minimum reward in $\breve{R}$ from all rewards. This is not a limitation of DM-IRL, but simply allows for a more convenient evaluation of learned rewards.

[5]The scoring scheme was as follows: patches dominated by road have a reward of 10, grass a reward of 1, dirt a reward of 0, shrubs a reward of -5, and trees a reward of -10. States with a mixture of elements received an approximate average by area.

Small Feature Pool We compute 5 features for each state, with each feature corresponding to a terrain type of either grass, tree, road, dirt, or shrub. We assign values to these features by manually segmenting small portions of the terrain corresponding to each feature type and then fitting a mixture of Gaussians with two components each for each terrain type to the pixel colors in YCbCr color space in the segment. By combining the outputs of all 5 mixture models, we obtain the probability that a given pixel in the image belongs to each of the 5 terrain types. We denote these probabilities the *pixel features*. The features for each state are the mean of the pixel features across pixels in that state. Non-regularized DM-IRL was used for this experement.

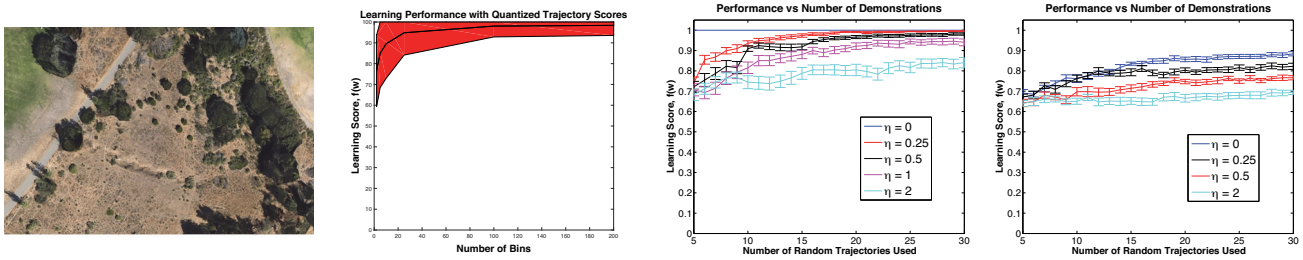
Figure 1(b) illustrates DM-IRL performance when the expert is forced to quantize scores. This can occur when a judge is asked to rate on a fixed scale. Learning utilized 30 randomly produced trajectories, as in the experiment for Figure 1(c). While performance decreases as the number of scoring bins shrinks, even with only 5 bins DM-IRL achieves a mean $f(\hat{\mathbf{w}})$ score of over 0.8.

Figure 1(c) shows the performance of DM-IRL on the satellite terrain domain with varying levels of expert scoring noise. Each trajectory was generated by selecting a start state uniformly at random and then executing π_{rand}, a policy that selects an action at random with uniform probability. Even with each trajectory score perturbed up to 100 percent ($\eta = 2$), DM-IRL produces a policy almost 0.8 times as good as optimal after observing only 9 demonstrations.

Expanded Feature Pool When the number of features is much larger, the problem is "trajectory-limited" and rank $(M) \ll k$, the dimensionality of the feature space. To explore this case, we use the same domain as above but with $k \approx 63,000$. Features consist of Gabor filter (Fogel and Sagi 1989) outputs (8 orientations and 5 scales), color histograms at multiple granularities, and 512 GIST features (Oliva and Torralba 2001). This overcomplete feature set illustrates a case where good features are not known *a priori*; we use this scenario to examine the Lasso varient of DM-IRL.

Figure 1(d) shows the results of regularized DM-IRL across varying levels of expert score noise. Because the learner had access to very few scores compared to number of features, the learner was not able to recover a perfect reward vector, even without expert noise. Nonetheless, the learner achieved $f(\hat{\mathbf{w}}) > 0.88$ after only 30 noiselessly scored demonstrations. With moderate amounts of noise in the expert scores, $\eta \leq 0.25$, the learner was still able to achieve good performance. As noise increases further performance begins to drop off noticeably.

Transfer Experiment We examine the ability of DM-IRL to transfer learning from demonstrators with unknown transition functions to a new MDP with novel states and transition function. We split the terrain from section 4.2 into two halves (Figure 2 (a), (b)). On the left half, demonstrations are acquired, in a round robin fashion, from three demonstrators executing a suboptimal policy under three differing transition functions. Demonstrator 1 becomes irrecoverably stuck in areas of trees with probability 1, demonstrator 2 can escape areas of trees with probability 0.5 at each attempt,



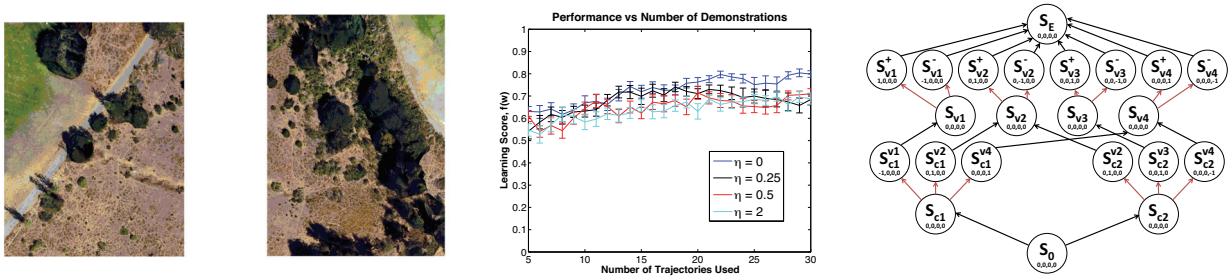
(a) Satellite Terrain (Google Earth)

(b) Quantized Performance

(c) Simple Features Performance

(d) Large Feature Pool Performance

Figure 1: a) Satellite Terrain. b) Learning performance over 10 trials $f(\hat{\mathbf{w}})$ for DM-IRL when $\breve{\mathbf{v}}$ is created by quantizing $\tilde{\mathbf{v}}$ into a finite number of evenly spaced bins; red area is 1 standard deviation. c) Learning performance, $f(\hat{\mathbf{w}})$, of DM-IRL with small feature set. d) Learning performance, $f(\hat{\mathbf{w}})$, of LASSO regularized DM-IRL with $\approx 63,000$ features. In c) and d), noise was added to the synthetic expert scores as $\breve{v}_i = \tilde{v}_i + \frac{1}{2}\mathcal{U}_i\tilde{v}_i \;\; \forall i \in \{1, ..., r\}$ where $\mathcal{U}_i$ is a random variable with value uniformly distributed between $-\eta$ and η. 20 trials were performed, bars indicate standard error.



(a) Demonstration Terrain

(b) Evaluation Terrain

(c) Transfer Performance

(d) 3-SAT Example

Figure 2: c) Transfer performance $f(\hat{\mathbf{w}})$ of LASSO regularized DM-IRL with $\approx 63,000$ features. Noise was added to the synthetic expert scores using the procedure from section 4.4.2. d) Example MDP for a 2 clause 3-SAT instance $(v_1 \vee \neg v_2 \vee \neg v_4) \wedge (\neg v_2 \vee \neg v_3 \vee v_4)$. Black transition arrows indicate no choice, multiple black arrows from a single state indicate uniform probability of taking any transition. Red transition arrows indicate a deterministic choice. $\phi(s)$ is indicated in each state.

and demonstrator 3 has a 0.5 probability to move in a random direction at every time step. Demonstrators have a uniform distribution over starting states, execute the true, optimal policy at each time step with probability 0.75, and take the action 90 degrees clockwise from optimal with probability 0.25. The learner's transition function is deterministic (allowing up, down, left, and right movement) and performance is evaluated on the right half of the terrain. This is a very challenging learning scenario with noisy trajectory scores, suboptimal demonstrations, an extremely small number of trajectories relative to the number of features, and requires transferring the learned reward to a new MDP. Even so, DM-IRL is performs well on the unseen right terrain with a reasonable number of trajectories (Figure 2 (c)).

Discussion and Conclusions

IRL approaches to LfD generally require a nearly optimal expert and often assume knowledge of the expert's transition function. DM-IRL bypasses these requirements by instead requiring an expert to score example trajectories, enabling learning from scored arbitrary trajectories in tasks that are difficult for humans to execute optimally. Unlike traditional approaches to IRL, the expert scorer need not actually perform the demonstrations, which may be generated by the learner or even a third party. This can significantly reduce the teaching burden on the expert (Cakmak and Thomaz 2012). Many tasks exist that are easy for humans to judge but difficult to perform, such as athletic performance or food preparation. By using an explicit scoring signal from the expert, DM-IRL does not require the demonstrator's transition function, which may be difficult to estimate. For instance, Internet or crowd sourced data sets could be scored by an expert and used to provide training data.

Given enough linearly independent trajectories, learned reward quality degrades gracefully as noise in the expert's scoring increases and, in practice, simple L_1 regularization can be an effective tool when the size of the feature space is much larger than the number of available trajectories. While DM-IRL is not a silver bullet for reward learning from demonstration, it is simple, fast, versatile, and is well suited for situations where traditional methods are not appropriate.

This material is based upon work supported by the NSF under

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *International Conference on Machine Learning*, volume 69, 1–8.

Akrour, R.; Schoenauer, M.; Sebag, M.; et al. 2013. Interactive robot education. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Workshop on Reinforcement Learning with Generalized Feedback: Beyond Numeric Rewards*.

Boularias, A.; Kober, J.; Peters, J.; and A. Boularias, J. K. 2011. Relative Entropy Inverse Reinforcement Learning. *International Conference on Automated Planning and Scheduling* 15:20 – 27.

Cakmak, M., and Thomaz, A. L. 2012. Designing robot learners that ask good questions. In *International Conference on Human-Robot Interaction*, 17–24.

Chatterjee, K.; Majumdar, R.; and Henzinger, T. A. 2006. Markov decision processes with multiple objectives. In *Symposium on Theoretical Aspects of Computer Science*. 325–336.

Daniel, C.; Viering, M.; Metz, J.; Kroemer, O.; and Peters, J. 2014. Active reward learning. In *Proceedings of Robotics: Science and Systems*.

El Asri, L.; Laroche, R.; and Pietquin, O. 2013. Reward shaping for statistical optimisation of dialogue management. In *Statistical Language and Speech Processing*. 93–101.

Fogel, I., and Sagi, D. 1989. Gabor filters as texture discriminator. *Biological Cybernetics* 61(2):103–113.

Karp, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. 85–103.

Kolter, J. Z.; Abbeel, P.; and Ng, A. Y. 2007. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Advances in Neural Information Processing Systems*, 769–776.

Levine, S.; Popovic, Z.; and Koltun, V. 2011. Nonlinear inverse reinforcement learning with gaussian processes. In Shawe-taylor, J.; Zemel, R.; Bartlett, P.; Pereira, F.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 24*. 19–27.

Lopes, M.; Melo, F.; and Montesano, L. 2009. Active learning for reward estimation in inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*. 31–46.

Ng, A. Y., and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, 663–670.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, 278–287.

Grant No. IIS-1208245. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision* 42:145–175.

Ramachandran, D., and Amir, E. 2007. Bayesian inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2586–2591.

Ratliff, N. D.; Bagnell, J. A.; and Zinkevich, M. A. 2006. Maximum margin planning. In *International Conference on Machine Learning*, 729–736.

Ratliff, N.; Bradley, D.; Bagnell, J. A.; and Chestnutt, J. 2007. Boosting structured prediction for imitation learning. In *Advances in Neural Information Processing Systems*.

Syed, U., and Schapire, R. E. 2007. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems*, 1449–1456.

Syed, U.; Bowling, M.; and Schapire, R. E. 2008. Apprenticeship learning using linear programming. In *International Conference on Machine Learning*, 1032–1039.

Tibshirani, R. 1996. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Association for the Advancement of Artificial Intelligence National Conference on Artificial Intelligence*, 1433–1438.