Inverse Reinforcement Learning from Experts with State-Dependent Suboptimality

Anthony DiGiovanni

November 19, 2020

Abstract

When the reward function of a Markov decision process cannot be easily specified by hand, inverse reinforcement learning (IRL) may be used to infer the reward from demonstrations by experts. Relaxing the assumption in existing IRL work of near-optimal demonstrations, I propose an algorithm for learning a reward function from suboptimal experts. This work constructs a model in which each expert takes actions with probability proportional to expected next-step reward times a latent variable β from a distribution conditioned on the state. MEIRL-EM computes point estimates of the reward function parameters by maximizing an importance-sampling approximation of the evidence lower bound. In simulations, algorithms based on max-likelihood on models of β as deterministic or uniform across states outperform MEIRL-EM, achieving total reward near that of a policy trained on the true reward function for several MDPs. Each algorithm's performance is robust to data simulated from experts whose policies follow the optimal action-value function Q^* rather than next-step reward.

1 Introduction

Inverse reinforcement learning (IRL) algorithms aim to infer the reward function of a Markov decision process (MDP) navigated by agents (experts) [NR00]. This approach is useful when a hard-coded reward function may result in unintended behavior, diverging from the goals of the designer [AOS$^+$16]. Some IRL applications include autonomous helicopter control [ACQN07] and robots that respect human physical boundaries [KP16]. Typically, the parameters of the reward function are estimated by maximum likelihood on some model of the expert state-action sequences (trajectories), assuming the experts' policies are approximately optimal [AD18]. This assumption may not be satisfied for realistic problems. Consider designing self-driving cars: while a reward function producing safe behavior is hard to design by hand, lending this problem well to IRL, a given expert may not drive optimally in all states. However, a group of multiple drivers with expertise in different states may collectively cover the state space with near-optimal demonstrations. To learn a reward function from data of this form, this work constructs algorithms for the setting in which each expert's probability of taking a given action in a state scales with an expert-specific function of the state.

2 Problem description and solution approach

Consider an episodic MDP $(\mathcal{S}, \mathcal{A}, \mathcal{T}, T, r, \rho)$, a tuple of, respectively: the state and action spaces, transition kernel, episode length, reward function, and initial state distribution. The objective is to learn a policy $\pi^* = \text{argmax}_\pi \, \mathbb{E}_{\pi,\mathcal{T}}(\sum_{t=1}^{T} r(s_t, a_t, s_{t+1}))$ maximizing expected cumulative reward. Assume $\mathcal{T}$ and ρ are known or accurately estimated. While r cannot be queried in training, we model r as a parametric r_θ, for unknown $\theta \in \mathbf{R}^d$. Thus finding the optimal policy requires the intermediate step of estimating θ. For experts $i = 1, ..., m$, we are given trajectories $\{\tau_i^{(n)}\}_{n=1}^N$ where $\tau_i^{(n)} = \{s_{i,1}^{(n)}, a_{i,1}^{(n)}, \ldots, s_{i,T_i}^{(n)}, a_{i,T_i}^{(n)}\}$. The sample index n is omitted for cleanliness of notation except where necessary.

Similar to [FLL17], we model the probability of each τ_i as proportional to its cumulative reward, weighted by terms $\beta_{i,t}$ representing the expert's degree of local optimality in each state: $\pi_i(a_{i,t}|s_{i,t}) \to \mathbf{1}\{a_{i,t} = \text{argmax}_a \, \mathbb{E}_{\mathcal{T}(s'|s_{i,t},a)}(r_\theta(s_{i,t}, a, s'))\}$ as $\beta_{i,t} \to \infty$. Under this model, at each time step, the expert i in state $s_{i,t}$ takes an action according to the policy:

$$\pi_i(a_{i,t}|s_{i,t}, \beta_{i,t}) \propto \exp[\beta_{i,t}\mathbb{E}_{\mathcal{T}(s'|s_{i,t},a_{i,t})}(r_\theta(s_{i,t}, a_{i,t}, s'))] \tag{1}$$

Expected next-step reward is used rather than the optimal action-value function Q^* to avoid non-differentiability when r_θ is non-Lipschitz [NS07], and may be more appropriate for myopic experts. As the objective is to learn r from demonstrators with varying degrees of optimality across states, we model each $\beta_{i,t}$ as an independent draw from $N(\mu_i(s_{i,t}), \sigma_i^2)$, where $\mu_i(s) = \alpha_i^T \eta(s)$ for unknown $\alpha_i \in \mathbf{R}_+^p$ and known $\eta : \mathcal{S} \to \mathbf{R}^p$ shared by all experts. To simplify notation, let $\mathcal{T}(s_{i,1}|s_{i,0}, a_{i,0}) = \rho(s_{i,1})$, φ be the $N(0,1)$ density, Z_i be the product of normalization constants for the sequence of $\pi_i(a_{i,t}|s_{i,t}, \beta_{i,t})$, and $R_{\theta,i,t} = \mathbb{E}_{\mathcal{T}(s'|s_{i,t},a_{i,t})}(r_\theta(s_{i,t}, a_{i,t}, s'))$. The trajectory density is $p_{\theta,\alpha_i,\sigma_i^2}(\tau_i) = \int_{\beta_i} p_{\theta,\alpha_i,\sigma_i^2}(\tau_i, \beta_i) d\beta_i$, where, based on the factorization from the Markovian properties of this model:

$$p_{\theta,\alpha_i,\sigma_i^2}(\tau_i, \beta_i) = \frac{1}{Z_i(\theta, \beta_i)} \prod_{t=1}^{T_i} \frac{1}{\sigma_i} \varphi\left(\frac{\beta_{i,t} - \mu_i(s_{i,t})}{\sigma_i}\right) \mathcal{T}(s_{i,t}|s_{i,t-1}, a_{i,t-1}) \exp(\beta_{i,t} R_{\theta,i,t})$$

Finding $\hat{\theta}$ maximizing the likelihood $\prod_{i=1}^m \prod_{n=1}^N p_{\theta,\alpha_i,\sigma_i^2}(\tau_i^{(n)})$ is challenging because the integral over β_i is in general intractable, as is Z_i for infinite $\mathcal{A}$. The first challenge motivates approximation via the evidence lower bound. For a distribution q_{ϕ_i}, we have [KW13]:

$$\log p_{\theta,\alpha_i,\sigma_i^2}(\tau_i) = D_{KL}(q_{\phi_i}(\beta_i) \,||\, p_{\theta,\alpha_i,\sigma_i^2}(\beta_i|\tau_i)) + \mathbb{E}_{q_{\phi_i}(\beta_i)}[\log p_{\theta,\alpha_i,\sigma_i^2}(\tau_i, \beta_i) - \log q_{\phi_i}(\beta_i)]$$

The KL-divergence term vanishes when $q_{\phi_i} = p_{\theta,\alpha_i,\sigma_i^2}(\beta_i|\tau_i)$. However, exact computation of this conditional density would require integration over β, which this decomposition was designed to avoid. Examining the form of the intractable conditional density:

$$p_{\theta,\alpha_i,\sigma_i^2}(\beta_i|\tau_i) = \frac{\frac{1}{Z_i(\theta,\beta_i)} \exp\left\{-\frac{1}{2\sigma_i^2}\sum_{t=1}^{T_i}[\beta_{i,t} - \left(\sigma_i^2 R_{\theta,i,t} + \alpha_i^T \eta(s_{i,t})\right)]^2\right\}}{\int \frac{1}{Z_i(\theta,\tilde{\beta})} \exp\left\{-\frac{1}{2\sigma_i^2}\sum_{t=1}^{T_i}[\tilde{\beta}_t - (\sigma_i^2 R_{\theta,i,t} + \alpha_i^T \eta(s_{i,t}))]^2\right\} d\tilde{\beta}}$$

Define $x_i = (\phi, \theta, \alpha_i, \sigma_i^2)$. Since $Z_i(\theta, \beta_i) = \prod_{t=1}^{T_i} \int_{\mathcal{A}} e^{\beta_{i,t}\mathbb{E}_{\mathcal{T}(s'|s_{i,t},a)}(r_\theta(s_{i,t},a,s'))} da$, the $\beta_{i,1}, ..., \beta_{i,T_i}$

are independent conditional on τ_i. Motivated by this independence and the Gaussian kernel in the numerator, let q_{x_i} be $N_{T_i}(\sigma_i^2 R_{\theta,i} + E_i^T\alpha_i + \phi_{i,1}\mathbf{1}, (\sigma_i^2 + \phi_{i,2})I_{T_i})$, where $R_{\theta,i}$ is the vector of $R_{\theta,i,t}$ and E_i is the matrix of vectors $\eta(s_{i,t})$. Then, by maximizing the evidence lower bound $\mathcal{L}(x_i;\tau_i) = \mathbb{E}_{q_{x_i}}[\log p_{\theta,\alpha_i,\sigma_i^2}(\tau_i,\beta_i) - \log q_{x_i}(\beta_i|\tau_i)]$ over x_i, we can approximately optimize $p_{\theta,\alpha_i,\sigma_i^2}(\tau_i)$. With the reparameterization trick [KW13], we reduce the variance of gradient approximations: since $\beta_i = \sigma_i^2 R_{\theta,i} + E_i^T\alpha_i + \phi_{i,1}\mathbf{1} + (\sigma_i^2+\phi_{i,2})^{1/2}\epsilon$, where $\epsilon \sim N(0, I_{T_i})$:

$$\begin{aligned}\nabla_x \mathcal{L}(x;\tau_i) = \mathbb{E}_\epsilon[&\nabla_x \log p_{\theta,\alpha_i,\sigma_i^2}(\tau_i, \sigma_i^2 R_{\theta,i} + E_i^T\alpha_i + \phi_{i,1}\mathbf{1} + (\sigma_i^2 + \phi_{i,2})^{1/2}\epsilon) \\ &- \nabla_x \log q_x(\sigma_i^2 R_{\theta,i} + E_i^T\alpha_i + \phi_{i,1}\mathbf{1} + (\sigma_i^2 + \phi_{i,2})^{1/2}\epsilon|\tau_i)]\end{aligned}$$

We approximate $\mathbb{E}_\epsilon\left(\frac{\partial}{\partial\theta}\log Z_i(\theta,\beta_i)\right)$ using importance sampling, following [FLA16]. Suppose M samples of actions are drawn uniformly from $\mathcal{A}$. By the Strong Law of Large Numbers $\frac{1}{M}\sum_{j=1}^M \mathrm{vol}(\mathcal{A})\exp(\beta_{i,t}\mathbb{E}_{\mathcal{T}(s'|s_{i,t},a_j)}(r_\theta(s_{i,t},a_j,s'))) \xrightarrow{a.s.} \int_{\mathcal{A}} \exp(\beta_{i,t}\mathbb{E}_{\mathcal{T}(s'|s_{i,t},a)}(r_\theta(s_{i,t},a,s')))da$. Then the differentiable approximation of $\log Z_i(\theta,\beta_i)$ is:

$$\tilde{L}_i(a_1,...,a_M;\theta,\beta_i) = \sum_{t=1}^{T_i}\log\left(\frac{1}{M}\sum_{j=1}^{M}\mathrm{vol}(\mathcal{A})\exp(\beta_{i,t}\mathbb{E}_{\mathcal{T}(s'|s_{i,t},a_j)}(r_\theta(s_{i,t},a_j,s')))\right)$$

The algorithm I propose for estimation of $(\theta,\alpha_i,\sigma_i^2,\phi_i)$, using gradient ascent on the differentiable approximation of $\mathcal{L}$, is summarized in Algorithm 1 (MEIRL-EM). Note that while MEIRL-EM inherits its structure from AEVB in [KW13], it is used to obtain point estimates rather than a Bayesian posterior, and "EM" in the name refers to its optimization of the evidence lower bound rather than to an exact resemblance to the EM algorithm.

Algorithm 1 EM Multi-expert Inverse Reinforcement Learning (MEIRL-EM)

```
 1: Init $x^{(0)}$; $a_1,...,a_M \sim \mathrm{Unif}(\mathcal{A})$; $\epsilon_1,...,\epsilon_B \sim N(0,I_{T_i})$; $\gamma^{(0)} \leftarrow 1$; $y^{(0)} \leftarrow x^{(0)}$; $\delta \leftarrow 0.5$
 2: for $k = 1,...,NK$ do
 3:     $\gamma^{(k+1)} \leftarrow \frac{1}{2}(1+\sqrt{1+4\gamma^{(k)2}})$
 4:     Approximate $\log\tilde{p}_{\theta,\alpha_i,\sigma_i^2}(\tau,\beta)$ using $\tilde{L}_i(a_1,...,a_M;\theta,\beta)$
 5:     $\mathbf{g} \leftarrow \frac{1}{B}\sum_{b=1}^{B}\sum_{t=1}^{T_i}\sum_{i=1}^{m}\nabla_x[\log\tilde{p}_{y^{(k)}}(\tau_i^{(k)},\beta_i^{(k)}(\epsilon_b)) - \log q_{y^{(k)}}(\beta_i^{(k)}(\epsilon_b)|\tau_i^{(k)})]$
 6:     $\mathbf{g} \leftarrow \frac{\mathbf{g}}{||\mathbf{g}||_2}$
 7:     $x^{(k+1)} \leftarrow y^{(k)} + \delta\mathbf{g}$
 8:     $y^{(k+1)} \leftarrow P_{\mathcal{C}}\left(x^{(k+1)} + \frac{\gamma^{(k)}-1}{\gamma^{(k+1)}}(x^{(k+1)} - x^{(k)})\right)$
 9:     $\delta \leftarrow 0.99\delta$
10:     if $\sum_{i=1}^m \mathcal{L}(x^{(k+1)};\tau_i^{(k)}) < \sum_{i=1}^m \mathcal{L}(x^{(k)};\tau_i^{(k)})$ then
11:         $y^{(k)} \leftarrow x^{(k)}$; $\gamma^{(k)} \leftarrow 1$
```

3 Methods and results

Because of feasibility restrictions in this model, MEIRL-EM projects iterates onto $\mathcal{C} = \{x \mid \sigma_i^2 > 0, \phi_{i,2} > 0, \alpha_{i,j} \geq 0\}$. To evaluate the sensitivity of accurate inference to correct specification of the expert models, this algorithm is compared with two benchmarks: MEIRL-Uniform and MEIRL-Deterministic, based on models where each β_i is constant across states or $\beta_{i,t} = \alpha_i^T \eta(s_{i,t})$ with no noise, respectively. These optimize the log-likelihood over (θ, β) or (θ, α), respectively. Otherwise, they are implemented identically to Algorithm 1 (SGD, acceleration, importance sampling, and projection onto $\mathcal{C}$) except only MEIRL-EM uses adaptive restart, since empirically it was found only to improve performance of MEIRL-EM. These algorithms were evaluated on MDPs based on 10 different random seeds as follows.

All MDPs were 16x16 grids, with an action space of the 4 cardinal directions. An agent moves in the intended direction (or stays in place if moving into a boundary) with probability 0.95, else in each direction with equal probability. The reward function was $\theta_0 + \theta^T \psi(s)$ for $\theta \in \mathbf{R}^8$ and radial basis functions (RBFs) ψ, where θ and the centers of ψ vary across seeds. Four experts shared an intercept $\mu^{(0)}$ and RBFs η with centers determining the locations of expert near-optimality, with a hyperparameter ω controlling the radius of the RBFs (Fig. 1). To evaluate robustness to model misspecification, data were generated from the policy:

$$\pi_i(a_{i,t} | s_{i,t}, \beta_{i,t}) \propto \exp[\beta_{i,t} Q^*_\theta(s_{i,t}, a_{i,t})] \tag{2}$$

For each MDP, evaluations were run for varying values of the hyperparameters N, σ_i^2, ω, and $\mu^{(0)}$. These were tests of performance as a function of, respectively: sample size, noise in the sampling of β, coverage of the state space by near-optimal demonstrations, and presence of anti-optimal demonstrations (negative β). An evaluation is defined as follows: N trajectories of length $T_i = 20$ were simulated from each expert according to the myopic policy (1) and Boltzmann rational policy (2). Given a set of 10 initial states S, the optimal policy was solved on the true reward by value iteration (with $\gamma = 0.9$), and total reward was recorded from episodes of length $T = 50$ starting at each state in S. For J trials each, the three algorithms computed estimates of θ from $K = 5$ shuffled epochs of the N trajectories, and total reward was recorded from T-episodes on S for policies trained to optimality with respect to each of these values of θ, as well as a random θ drawn from the same distribution as the initial θ.

On average over all MDPs and hyperparameter combinations, none of the algorithms matched the true reward (Fig. 2a). However, under default hyperparameters $N = 100$, $\omega = 0.01$, $\sigma_i^2 = 0.1$, and $\mu^{(0)} = 0$ (these values ensure a signal-to-noise ratio in the expert data that makes the learning task nontrivial yet feasible), Deterministic and Uniform come moderately close to the true reward when trained on myopic expert data, and nearly match the true reward given Boltzmann data (Fig. 2b,c). The latter is the only setting where MEIRL-EM approaches the true reward performance. While the model underlying MEIRL-EM is best specified for the data, its weaker performance may be a result of the gap between optimizing the evidence lower bound versus the true likelihood. The family of the conditional density q_ϕ may not be sufficiently expressive to converge to the true density. More surprising is the strong performance of MEIRL-Uniform, which has unclear causes but may be due to a strong decrease in variance of estimates based on a biased yet simpler model.

Based on results for the separate seeds, each algorithm's performance varies depending on the MDP structure. With myopic data, all algorithms struggle in MDPs where there is only one hub of high-reward states amid low-reward states (Fig. 3a, 4a). Other MDPs where

(a) (b)

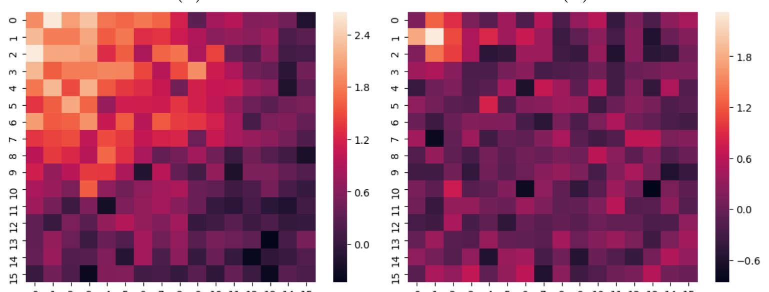

Figure 1: Plots of $\mu_1(s) + N(0, \sigma_1^2)$ for every s with $\sigma_1^2 = 0.1$, when (a) $\omega = 0.01$ versus (b) 0.5.

(a) (b) (c)

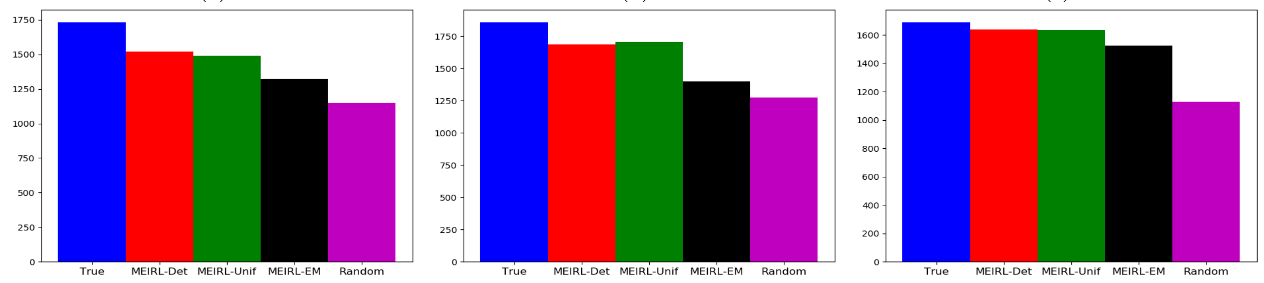

Figure 2: Mean cumulative reward across all MDPs, for all hyperparameters (left); and for default hyperparameters with myopic policy data (center) and Boltzmann policy data (right).

high-reward states are more common have more consistent success, yet note the much lower reward achieved with random θ, implying these MDPs are nontrivial (Fig. 3b, 4a).

Training on Boltzmann rational data consistently improves performance for all algorithms (Fig. 4a). It is unclear why this occurs; while the demonstrations from experts following policy (2) have stronger "signal," i.e. experts in states of near-optimality will consistently seek the highest-reward states, the algorithms use incorrect models. When $\mu^{(0)}$ is negative, causing negative β on average for states of low expertise, all algorithms do worse, however Deterministic is most robust to this change (Fig. 4b). Intuitively, when β may be negative, this permits systematic demonstrator errors: the algorithms struggle to distinguish a state of high reward successfully pursued by an expert with positive β, from one with low reward *mistakenly* pursued by an anti-optimizing expert with negative β. Performance also decreases strongly when the basis functions η are such that the experts are near-optimal in almost none of the states, as expected, since this change removes most of the signal from the demonstrations (Fig. 4c). In most MDPs, the algorithms are robust to increases of σ_i^2, i.e. noisy β (Fig. 4d). Finally, robustness to decreased sample size varies across MDPs (Fig. 4e).

	MEIRL-Det	MEIRL-Unif	MEIRL-EM
Mean Runtime (sec)	3.552	3.739	22.94

Table 1: Mean (across 100 repetitions) time for each algorithm to compute an estimate of θ.

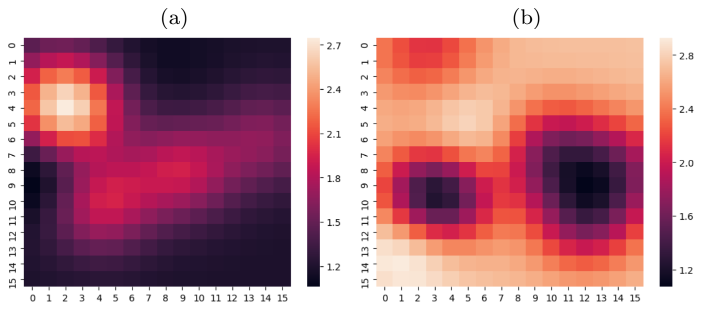

Figure 3: Example MDPs for seeds (a) 140 and (b) 180.

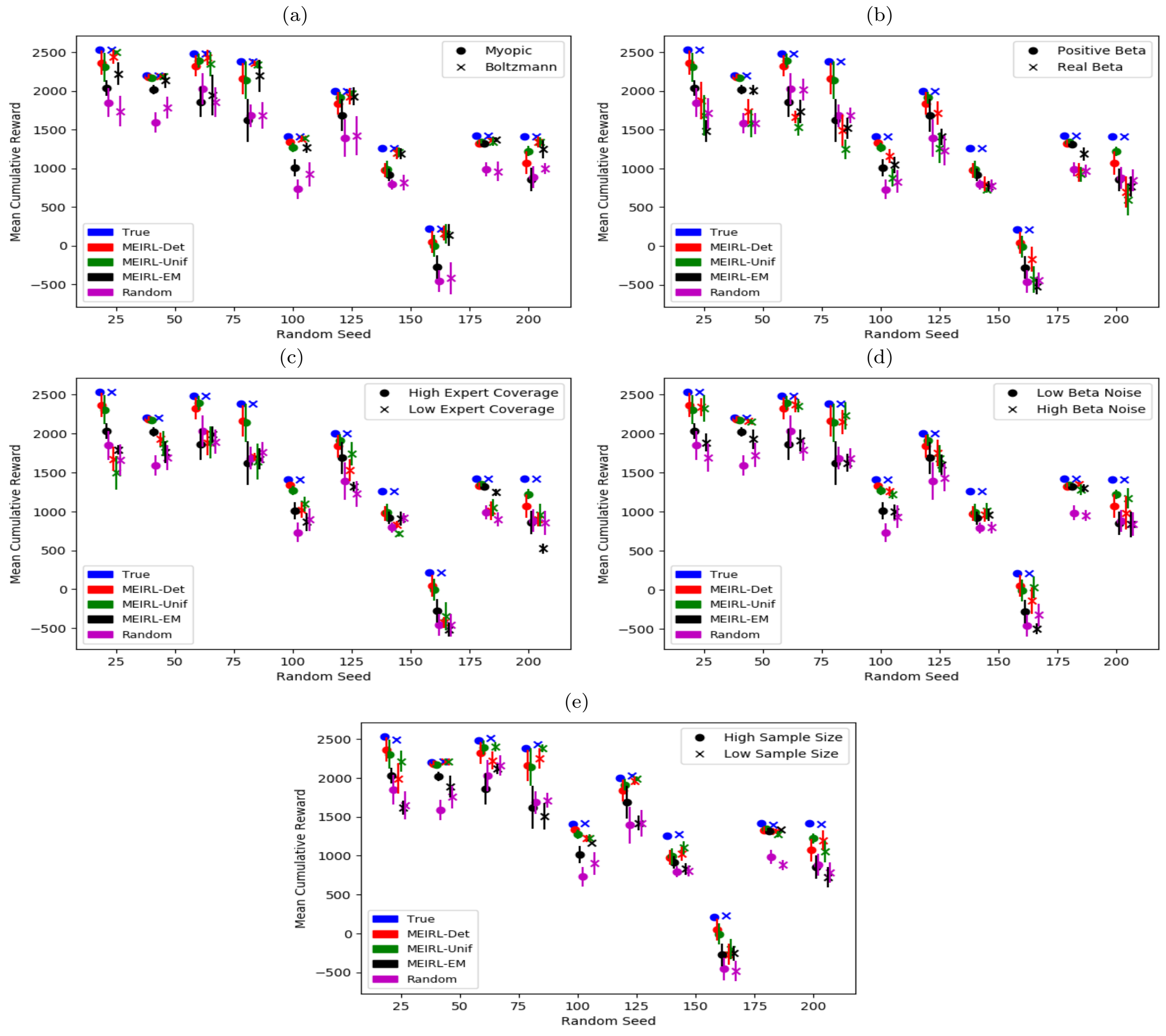

Figure 4: Mean cumulative reward for each MDP, comparing (a) data from policies (1) vs. (2); (b) $\mu^{(0)} = 0$ vs. -1; (c) $\omega = 0.01$ vs. 0.5; (d) $\sigma_i^2 = 0.1$ vs. 5; and (e) $N = 100$ vs. 20. Error bars are 2 standard errors. (Jitter is used to make the differences between algorithms visible for 10 seeds.)

References

[ACQN07] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1–8. MIT Press, 2007.

[AD18] S. Arora and P. Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *arXiv preprint arXiv:1806.06877*, 2018.

[AOS$^+$16] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[FLA16] C. Finn, S. Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. *arXiv preprint arXiv:1603.00448*, 2016.

[FLL17] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. *CoRR*, abs/1710.11248, 2017.

[KP16] B. Kim and J. Pineau. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, 8:51–66, 01 2016.

[KW13] D. P. Kingma and M. Welling. Auto-encoding variational bayes. 2013.

[NR00] A. Ng and S. Russell. Algorithms for inverse reinforcement learning. *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670, 2000.

[NS07] G. Neu and C. Szepesvari. Apprenticeship learning using inverse reinforcement learning and gradient methods. *Proc. UAI*, page 295–302, 2007.