2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Gradient–Based Inverse Risk-Sensitive Reinforcement Learning

Eric Mazumdar, Lillian J. Ratliff, Tanner Fiez, and S. Shankar Sastry

Abstract— **We address the problem of *inverse reinforcement learning* in Markov decision processes where the agent is *risk-sensitive*. In particular, we model risk-sensitivity in a reinforcement learning framework by making use of models of human decision-making having their origins in behavioral psychology and economics. We propose a gradient-based inverse reinforcement learning algorithm that minimizes a loss function defined on the observed behavior. We demonstrate the performance of the proposed technique on two examples, the first of which is the canonical *Grid World* example and the second of which is an MDP modeling passengers' decisions regarding ride-sharing. In the latter, we use pricing and travel time data from a ride-sharing company to construct the transition probabilities and rewards of the MDP.**

I. Introduction

The modeling and learning of human behaviors is becoming increasingly important as critical systems begin to rely more on automation and artificial intelligence. Yet, in this task we face a number of challenges, not least of which is the fact that humans are known to behave in ways that are not completely rational. For example, there is mounting evidence to support the fact that humans often use *reference points*—experiences that are perceived to be related to the decision the human is making. It has also been observed that their decisions are impacted by their perception of the external world (exogenous factors) and their present state of mind (endogenous factors) as well as how the decision is *framed* or presented [1], [2].

The success of *descriptive* behavioral models in capturing human behavior has long been touted by the psychology community and, more recently, by the economics community. Meanwhile, in the engineering context, humans have largely been modeled, under rationality assumptions, from the so-called *normative* point of view where things are modeled *as they ought to be*, which is counter to a descriptive *as is* point of view. However, risk-sensitivity has been fairly extensively explored in engineering and computer science in the context of learning to control stochastic dynamic systems (see, *e.g.*, [3], [4]). Many of these approaches are targeted at mitigating risks due to uncertainties in controlling a system such as a plant or robot. Much of this work simply handles *risk-aversion* by leveraging techniques such as exponential utility functions or minimizing variance. Human decision makers, on the other hand, can be at once risk-averse and risk-seeking depending their frame of reference. The complex risk-sensitive behavior arising from human interaction with automation is only recently coming into focus. Indeed, the adoption of diverse behavioral models in engineering—in particular, in learning and control—is growing due to the fact that humans are increasingly playing an integral role in automation both at the individual and societal scale.

The problem of learning accurate models of human decision-making is becoming increasingly important for both *prediction* and *description*. For example, control/incentive schemes need to predict human behavior as a function of external stimuli. On the other hand, policy makers, *e.g.*, are interested in interpreting human reactions to implemented regulations and policies. A few approaches for integrating the risk-sensitivity of humans in the control and reinforcement learning problems via behavioral models [5]–[8] have recently emerged. These approaches largely assume a risk-sensitive Markov decision process (MDP) formulated based on a behavioral model and determine the optimal policy via, *e.g.*, a learning procedure. We refer to this as the *forward* problem. We are interested in solving the so-called *inverse* problem which seeks to estimate the decision-making model given a set of demonstrations.

In particular, in this paper we model human decision-makers as *risk-sensitive Q-learning agents* where we exploit very rich behavioral models from behavioral psychology and economics that capture a whole spectrum of risk-sensitivity. We propose a gradient-based learning algorithm for inferring the decision-making model parameters from demonstrations—that is, we propose a novel framework for solving the *inverse risk-sensitive reinforcement learning* problem. We demonstrate the efficacy of the learning scheme on i) the canonical Grid World example and ii) a passenger's view of ride-sharing as an MDP with model parameters estimated from real-world data.

The remainder of this paper is organized as follows. In Section II, we briefly overview the Q-learning model we assume for risk-sensitive agents and show that it is amenable to integration with the behavioral models. In Section III, we formulate the inverse risk sensitive RL problem and propose a gradient–based algorithm to solve it. Examples that demonstrate the ability of the proposed scheme to capture a wide breadth of risk-sensitive behaviors are provided in Section IV. Finally, we conclude with some discussion and comments on future work in Section V.

E. Mazumdar and S. S. Sastry are with the Electrical Engineering and Computer Sciences Department at the University of California, Berkeley. email: {mazumdar, sastry}@eecs.berkeley.edu

T. Fiez and L. Ratliff are with the Electrical Engineering Department at the University of Washington, Seattle. email: {fiezt, ratliffl}@uw.edu

This work is supported by NSF CRII Award CNS-1656873, NSF US-Ignite Award CNS-1646912, NSF FORCES (Foundations Of Resilient CybEr-physical Systems) Award CNS-1238959, CNS-1238962, CNS-1239054, CNS-1239166"

II. Risk-Sensitive Reinforcement Learning

Risk-sensitivity has been long covered in the stochastic decision-making literature. The typical way in which risk is dealt with is through transformations of the reward by exponential value functions. As an alternative approach, transformation of the *temporal differences* in a Q-learning scheme according to a risk model has recently been proposed [6]–[8]. This avoids certain pitfalls of the reward transformation approach such as poor convergence performance. Further, this framework allows for integration of behavioral decision-making models [5]–[7].

Under the assumption that the agent of interest is making decisions according to this model, we formulate a gradient–based method for learning the parameters of the agent's value function from data. In this section, we briefly review relevant features of risk-sensitive Q-learning. A more thorough treatment of the subject can be found in [6]–[8].

A. Markov Decision Process

We consider a class of finite MDPs which consist of a state space X, an admissible action space $A(x) \subset A$ for each $x \in X$, a transition kernel $P(x'|x,a)$ (which denotes the probability of moving from state x to x' given action a), and a reward function $r : X \times A \times W \to \mathbb{R}$ where W is the disturbance space and has distribution $P_r(\cdot|x,a)$. Including disturbances w allows us to model random rewards for which use the notation $R(x,a)$ to denote the random reward having distribution $P_r(\cdot|x,a)$.

In the classical expected utility maximization framework, an agent with utility function u seeks to maximize the expected discounted utility by selecting a Markov policy π. That is, for an infinite horizon MDP, the optimal policy is obtained by maximizing

$$J(x_0) = \max_\pi \mathbb{E}\left[\textstyle\sum_{t=1}^{\infty} \gamma^t u(R(x_t,a_t))\right] \tag{1}$$

where x_0 is the initial state and $\gamma \in [0,1)$ is the discount factor. The risk-sensitive RL problem transforms the above problem to account for a salient features of human decision-making such as loss aversion, reference point dependence, and framing effects.

B. Value Functions

In the risk-sensitive setting, just like the standard expected utility framework, an agent makes choices based on the value of their outcome as defined by their *value function* $u : \mathbb{R} \to \mathbb{R}$.

One particularly rich class of value functions is the parametric class of value functions introduced in Prospect Theory by Kahneman and Tvsersky [9], [10]. This class captures many of the most salient features of human decision-making including framing effects, reference dependence and loss aversion. The value function is given by:

$$u(y) = \begin{cases} c_+(y-y_0)^{\rho_+}, & y > y_0 \\ -c_-(y_0-y)^{\rho_-}, & y \le y_0 \end{cases} \tag{2}$$

where y_0 is the reference point that the decision-maker compares outcomes against when determining if the outcome is a loss or gain. The parameters $(c_+, c_-, \rho_+, \rho_-)$ control the degree of risk-sensitivity and loss-aversion, allowing the value function to capture a rich set of behaviors. For example, $0 < \rho_+, \rho_- < 1$ leads to risk-averse preferences on gains and risk-seeking preferences on losses (concave in gains, convex in losses) while $\rho_+, \rho_- > 1$ leads to risk-averse preferences on losses and risk-seeking preferences on gains (convex in gains, concave in losses).

We introduce a class of logarithm–based value functions that approximate the shape of the prospect theory value function for $\rho_+, \rho_- < 1$, the range of values that are typical of human decision-making [11]–[13].

$$u(y) = \begin{cases} c_+ \log(1+\rho_+(y-y_0)), & y > y_0 \\ -c_- \log(1+\rho_-(y_0-y)), & y \le y_0 \end{cases} \tag{3}$$

This new value function improves the performance (in terms of convergence speed) of the gradient-based IRL algorithm we propose in Section III.

Outside of the prospect theory value function, other mappings have been proposed to capture risk-sensitivity. One example is the entropic map, $u(y) = \exp(\lambda y)$, where the one parameter, λ controls the degree of risk-sensitivity. This value function has been primarily used in finance and control theory.

C. Valuation Functions

More complex value functions are not enough to capture risk-sensitive behavior. To further extend classical RL to risk-sensitive RL, *valuation functions* are used to generalize the expectation operator.

Definition 1 ([6], [14], [15]): A mapping $\mathscr{V} : \mathbb{R}^{|I|} \times \mathscr{P} \to \mathbb{R}$ is called a *valuation function* if for each $\mu \in \mathscr{P}$, (i) $\mathscr{V}(Y,\mu) \le \mathscr{V}(Z,\mu)$ whenever $Y \le Z$ (monotonic) and (ii) $\mathscr{V}(Y + y\mathbf{1}, \mu) = \mathscr{V}(Y,\mu) + y$ for any $y \in \mathbb{R}$ (translation invariant).
Such a map is used to characterize an agent's preferences—that is, one prefers (Y,μ) to (Z,ν) whenever $\mathscr{V}(Z,\nu) \le \mathscr{V}(Y,\mu)$.

In the context of MDPs, we can define a valuation function for each state–action pair. We refer to $\mathscr{V}(Y|x,a) : \mathbb{R}^{|I|} \times X \times A \to \mathbb{R}$ as a *valuation map* such that $\mathscr{V}_{x,a} \equiv \mathscr{V}(\cdot|x,a)$ is a valuation function.

If we let $\mathscr{V}_x^\pi(Y) = \sum_{a \in A(x)} \pi(a|x) \mathscr{V}_{x,a}(Y)$, the optimization problem in (1) generalizes to

$$\begin{aligned}\tilde{J}_T(\pi, x_0) =& \mathscr{V}_{x_0}^{\pi_0}\Big[R[x_0,a_0] + \gamma \mathscr{V}_{x_1}^{\pi_1}\big[R[x_1,a_1] + \\ & \cdots + \gamma \mathscr{V}_{x_T}^{\pi_T}[R(x_T,a_T)] \cdots\big]\Big].\end{aligned} \tag{4}$$

Let us define $\max_\pi \tilde{J}(\pi,x_0) = \lim_{T\to\infty} \tilde{J}_T(\pi,x_0)$.

The particular valuation function we employ in our forward model is the *shortfall valuation*, which incorporates reference dependence in the decision-making model. It originated in mathematical finance [14], [16] where it was shown to be a valuation function and it has been used in a risk-sensitive MDP context in [3] and in a RL context in [6]. The shortfall $\mathscr{V}$ induced by a value function $u : \mathbb{R} \to \mathbb{R}$ and an acceptance level r_0 is given by

$$\mathscr{V}(Y) = \sup\left\{m \in \mathbb{R} | \mathbb{E}_\mu\left[u(Y-m)\right] \ge r_0\right\} \tag{5}$$

where we suppress the dependence of $\mathscr{V}$ on μ, the distribution of the random variable for outcomes Y.

D. Risk-Sensitive Q-Learning

Given the value and valuation functions, we are now introduce risk-sensitive Q-learning. In the classical RL framework, the Bellman equation is used to derive a Q-learning procedure. Generalizations of the Bellman equation for risk-sensitive RL have been derived in [3], [7]. These generalizations are then used to formulate a Q-learning procedure for the risk-sensitive RL problem.

Given (4), it is shown in [3] that if V^* satisfies

$$V^*(x_0) = \max_{a\in A(x)} \mathscr{V}_{x,a}(R(x,a)+\gamma V^*), \tag{6}$$

then $V^* = \max_\pi \tilde{J}(\pi, x_0)$ holds for all $x_0 \in X$.

We now define the Q-function, $Q^*(x,a) = \mathscr{V}_{x,a}(R(x,a)+\gamma V^*)$, such that (6) becomes

$$Q^*(x,a) = \mathscr{V}_{x,a}\left(R(x,a)+\gamma \max_{a\in A(x')} Q^*(x',a)\right), \tag{7}$$

for all $(x,a)\in X\times A$.

If u is continuous and strictly increasing, we can evaluate (7) for the shortfall valuation, and get the following fixed point equation for Q^* (see [14, Prop. 4.104] or [6, Prop. 3.1]):

$$\mathbb{E}\left[u\left(r(x,a,w)+\gamma \max_{a'\in A(x')} Q^*(x',a') - Q^*(x,a)\right)\right] = r_0 \tag{8}$$

where the expectation is taken with respect to $\mu = P(x'|x,a)P_r(w|x,a)$. Further it is shown that in finite MDPs, and with further technical conditions on u, there exists a unique Q-function Q^* satisfying (8) for all $(x,a)\in X\times A$ [6, Thm. 3.2].

III. Gradient–Based Inverse Risk-Sensitive RL

We now assume that an agent has performed risk-sensitive Q-learning to obtain their policy and formulate the inverse risk-sensitive RL problem in a general form as follows:

First, we select a parametric class of policies, $\{\pi_\theta\}_\theta$, $\pi_\theta \in \Pi$ and parametric utility function $\{u_\theta\}_\theta$, $u_\theta \in \mathscr{F}$ where $\mathscr{F}$ is a family of utility functions and $\theta\in\Theta\subset\mathbb{R}^d$. We try to *tune* the parameters so as to minimize some loss $\ell(\pi_\theta;\mathscr{D})$ which is a function of the parameterized policy π_θ and a set of *demonstrations* given as state-action pairs, $\mathscr{D} = \{(x_k,a_k)\}_{k=1}^N$.

We will use the shorthand notation $\ell(\theta) = \ell(\pi_\theta;\mathscr{D})$, and for mappings u and Q, we will now indicate their dependence on θ—that is, we will write $Q(x,a,\theta)$ and $u(y,\theta)$ where $u: Y\times\Theta\to\mathbb{R}$. Note that since y is the temporal difference it also depends on θ and we will indicate that where not obvious by writing $y(\theta)$.

It is common in the IRL literature to adopt a smooth map G that operates on the space of Q-functions for defining the parametric policy space. A common class of parametric policies are Boltzmann policies of the form

$$G_\theta(Q)(a|x) = \frac{\exp(\beta Q(x,a,\theta))}{\sum_{a'\in A}\exp(\beta Q(x,a',\theta))} \tag{9}$$

where $\beta>0$ controls how close $G_\theta(Q)$ is to a *greedy policy* which we define to be any policy such that $\sum_{a\in A}\pi_\theta(a|x)Q(x,a,\theta) = \max_{a\in A} Q(x,a,\theta)$. We will utilize policies of this form.

We will use value functions such as those described in Section II-B. For example, if u is the prospect theory value function defined in (2), then the parameter vector is $\theta = (c_+, c_-, \rho_+, \rho_-, \gamma, \beta)$.

The optimization can be stated generally as

$$\min_{\theta\in\Theta}\{\ell(\theta)|\ \pi_\theta = G_\theta(Q^*), u_\theta\in\mathscr{F}\} \tag{10}$$

Given a set of *demonstrations* $\mathscr{D} = \{(x_k,a_k)\}_{k=1}^N$, it is our goal to recover the policy and estimate the value function.

A possible loss function is the negative log-likelihood of the demonstrated behavior which is given by

$$\ell(\theta) = \sum_{(x,a)\in\mathscr{D}} \tfrac{n(x,a)}{N}\log(\pi_\theta(x,a)) \tag{11}$$

where where $n(x,a)$ is the frequency of (x,a) in $\mathscr{D}$. An alternative to the log-likelihood is the relative entropy or KullbackLeibler (KL) divergence between the empirical distribution of the state-action trajectories and their distribution under the learned policy—that is, we could minimize

$$\ell(\theta) = \sum_{x\in\mathscr{D}_x} D_{\mathrm{KL}}(\hat{\pi}(\cdot|x)||\pi_\theta(\cdot|x)) \tag{12}$$

where $D_{\mathrm{KL}}(P||Q) = \sum_i P(i)\log(P(i)/Q(i))$, $\mathscr{D}_x\subset\mathscr{D}$ is the sequence of observed states and $\hat{\pi}$ is the empirical distribution on the trajectories of $\mathscr{D}$.

We propose to solve the problem of estimating the parameters of the agent's value function and approximating the agent's policy via gradient methods. This requires computing the derivative of π_θ with respect to θ_k, an element of $\theta\in\Theta$, which in turn requires computing the derivative of $Q^*(x,a,\theta)$ with respect to θ_k. Indeed, given the smooth map G, the dependence of $D_{\theta_k}\pi_\theta(a|x)$ on $D_{\theta_k}Q^*(x,a,\theta)$ is clear:

$$D_{\theta_k}\pi_\theta(a|x) = \pi_\theta(a|x)D_{\theta_k}\ln(\pi_\theta(a|x)) \tag{13}$$

$$= \pi_\theta(a|x)\beta(D_{\theta_k}Q^*(x,a,\theta)... \tag{14}$$

$$-\sum_{a'\in A}\pi_\theta(a'|x)D_{\theta_k}Q^*(x,a',\theta)) \tag{15}$$

Hence, given the form of the Q-learning procedure where the temporal differences are transformed as in [6], we need to derive a mechanism for obtaining the optimal Q, show that it is in fact differentiable, and derive a procedure for obtaining the derivative.

We will show that $D_{\theta_k}Q^*_\theta$ can be calculated almost everywhere on Θ by solving fixed-point equations similar to the Bellman-optimality equations.

We first describe our our assumptions on u.

Assumption 1: The value function $u: Y\times\Theta\to\mathbb{R}$ satisfies the following: (i) it is strictly increasing in y and for each $\theta\in\Theta$, there exists a y such that $u(y,\theta) = r_0$; (ii) for each $\theta\in\Theta$, it is Lipschitz in y with constant $L_y(\theta)$ and locally Lipschitz on Θ with constant L_θ; (iii) there exists $\varepsilon>0$ such that $\varepsilon\le\frac{u(y,\theta)-u(y',\theta)}{y-y'}$ for $y\neq y'$; and (iv) $u\in C^1(Y\times\Theta,\mathbb{R})$.

Define $L_y = \min_\theta L_y(\theta)$ and $L = \max_\theta\{L_y(\theta), L_\theta\}$. Let $\tilde{u}\equiv u - r_0$. We define the map T such that

$$(TQ)(x,a,\theta) = \alpha\mathbb{E}_{x,a,w}\tilde{u}(y(\theta),\theta) + Q(x,a,\theta) \tag{16}$$

where $y(\theta) = r(x,a,w) + \gamma \max_{a' \in A} Q(x',a,\theta) - Q(x,a,\theta)$.

Under our assumptions, this mapping is a contraction, with Q^* as its unique fixed point. For cases where $r_0 = 0$, this was first shown in [7] and in the more general setting using similar techniques in [6].

Given the map T, we now state our main theorem:

Theorem 1 ([17]): Assume that $u : Y \times \Theta \to \mathbb{R}$ satisfies Assumption 1. Then the following statements hold:

(a) Q^*_θ is locally Lipschitz-continuous as a function of θ—that is, for any $(x,a) \in X \times A$, $\theta, \theta' \in \Theta$, $|Q^*(x,a,\theta) - Q^*(x,a,\theta')| \leq R\|\theta - \theta'\|$ for some $R > 0$;

(b) Except on a set of measure zero, the gradient $D_\theta Q^*_\theta$ is given by the solution of the fixed–point equation

$$\begin{aligned}\phi_\theta(x,a) =& \alpha \mathbb{E}_{x,a,w}\big[D_\theta \tilde{u}(y(\theta),\theta) + D_y \tilde{u}(y(\theta),\theta) \\ &\cdot (\gamma \phi_\theta(x', a^*_{x'}) - \phi_\theta(x,a))\big] + \phi_\theta(x,a)\end{aligned} \quad (17)$$

where $\phi_\theta : X \times A \to \mathbb{R}^d$.

The proof of the above theorem is provided in our paper [17].

Theorem 1 gives us a procedure—namely, a fixed–point equation which is a contraction—to compute the derivative $D_{\theta_k} Q^*$ so that we can compute the derivative of our loss function $\ell(\theta)$. Hence the gradient method provided in Algorithm 1 for solving risk-sensitive IRL problem is well formulated.

Algorithm 1 Gradient-Based Inverse Risk-Sensitive RL

```
1: procedure RISKIRL($\mathscr{D}$)
2:     Initialize: $\theta \leftarrow \theta_0$
3:     while $k <$ MAXITER & $\|\ell(\theta) - \ell(\theta_-)\| \geq \delta$ do
4:         $\theta_- \leftarrow \theta$
5:         $\eta_k \leftarrow$ LINESEARCH($\ell(\theta_-), D_\theta \ell(\theta_-)$)
6:         $\theta \leftarrow \theta_- + \eta_k D_\theta \ell(\theta_-)$
7:         $k \leftarrow k+1$
8:     return $\theta$
```

Remark 1: The prospect theory u given in (2) is not globally Lipschitz in y—in particular, it is not Lipschitz near the reference point y_0—for values of ρ_+ and ρ_- in less than one. Moreover, for certain parameter combinations, it may not even be differentiable. In [6], the authors propose some techniques such as truncation for handling the RL problem numerically when u is not globally Lipschitz. This also motivates the use of the logarithm–based value function that well-approximates the prospect theory u and retains the convex-concave structure.

IV. EXAMPLES

We now demonstrate the proposed methods performance on two examples. While we are able to formulate the risk-sensitive IRL problem for parameter vectors θ that include γ and β, in the following examples we use $\gamma = 0.95$ and $\beta = 0.5$. Further, we fix the reference points of the log and prospect value functions to 0 and the acceptance level for the shortfall valuation to 0. The purpose of doing this is to explore the effects of changing only the parameters of the value function on the resulting policy. For both experiments, the loss function considered is the negative log-likelihood (11), and data is generated by sampling the given policy.[1]

A. Prospecting in Grid World

We tested Algorithm 1 on data from agents operating on the canonical Grid World MDP example shown in Fig. 1 where the agents all start in the blue box, aim to maximize their value function over an infinite horizon. In particular, each square represents a state, and the action space is $A = \{N, NE, E, SE, S, SW, W, NW\}$. Each action corresponds to a movement in the specified direction. With probability 0.93, the agent moves in the desired direction and with probability 0.01, they move in any of the other 7 directions. To make the grid finite, any action taking the agent out of the grid has probability zero, and the other actions are re-weighted accordingly. The black and hatched green states are absorbing and give rewards of -1 and $+1$, respectively. All other states give rewards of 0.1.

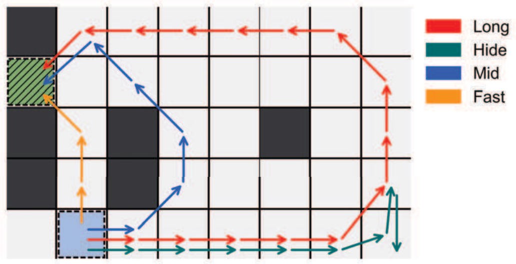


Fig. 1: True and Learned paths extracted from the optimal policies for agents with logarithmic value functions. The true and learned paths align exactly with one another. 'Hide','Long', and 'Mid' correspond to increasingly risk-averse agents while 'Fast' corresponds to a risk-neutral agent.

Fig. 1 provides a visualization of the agent's risk sensitivity resulting from different parameter combinations using the logarithm–based value function for the true and learned agents. There are four general classes of behaviors each having different risk/reward tradeoffs. Each path in Fig. 1 presents a different risk profile, with the 'Fast' path being the riskiest but potentially the most rewarding, and the 'Hide' path being the most conservative.

In our first experiment, we trained *true* agents with the prospect, entropic, and log value functions as described in Section II-B, with various sets of parameters to generate the four behaviors. We then collected 10,000 sample trajectories from these agents, and learned the parameters of the value functions using our gradient–based approach. We calculated the distance in the L_1 norm between the policy in state x of the true agent and the policy in state x of the learned agent. In Table I we report the mean distance across states, as well as the variance in the distance across states.

We first note that in all the cases considered in Table I, the learned value functions produce policies that correctly

[1]We note that learning the decision-making model is different than learning a reward function that can recreate the behavior (which is the classical IRL problem), thus, benchmarking our approach against standard IRL is not suitable.

	Prospect		Log		Entropic Map	
	Mean	Var	Mean	Var	Mean	Var
Fast	0.033	0.029	4.5e-3	5.1e-3	3e-4	6e-4
Mid	0.026	0.020	0.011	6.5e-3	9.3e-3	7.3e-3
Long	0.076	0.051	0.014	8.3e-3	1.7e-3	1.1e-3
Hide	0.144	0.011	0.050	0.043	9e-4	7e-4

TABLE I: Error mean and variance, as measured by the L_1 norm, between the learned and true behaviors for various degrees of risk-aversion in agents with different types of value functions. The 'Fast' agent is risk-neutral; and 'Mid', 'Long', and 'Hide' have increasing degree of risk-aversion. For the prospect value function agents the behaviors were generated by keeping c_+, ρ_-, ρ_+ fixed to: $c_+ = 1$ and $\rho_- = \rho_+ = 0.5$ and varying c_-. The parameter values for the log value function agents are described under Fig. 1. For the entropic map value function agents we varied lambda from 0.1 to -0.3 to generate the 4 behaviors.

match the maximum likelihood path of the true agent. We also note that performance for learning a prospect value function was markedly worse than learning entropic map or log value functions. This is most likely due to the fact that the prospect value function is not Lipschitz around the reference point. Thus, we have no guarantees of differentiability of Q^* with respect to θ, and the results are of worse quality. The entropic map performs best of the three value functions, mainly because there is only one parameter to learn, and the rewards and losses are all relatively small.

In our second experiment, we learned parameters for

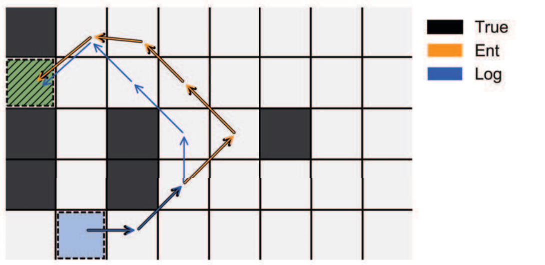


Fig. 2: Maximum Likelihood paths extracted from the optimal policies for agent's with a prospect value function learned with an entropic and logarithm value function. The true and learned paths coincide exactly for the true prospect and learned entropic value functions while the learned logarithmic value function over approximates how risk-averse the true agent is. The true prospect agent has parameters $(c_+, c_-, \rho_+, \rho_-) = (10, 1, 0.5, 0.5)$. The learned logarithmic agent has parameters $(c_+, c_-, \rho_+, \rho_-) = (4.6, 0.3, 1, 1.5)$ and the learned entropic agent has $\lambda = -0.13$.

agents operating under the entropic map and log value functions from data collected from an agent operating under the prospect value function and exhibiting the 'Mid' behavior. The maximum likelihood path of the true prospect agent and the learned entropic map and log value function agents is shown in Fig. 2. The mean L_1 distance between the true prospect agent and the learned agents is shown in Table II

Of note with the experiments on learning prospect agents with log and entropic agents is the fact that, though the log value function fails to capture the correct maximum likelihood path, it is closer, on average to the true policy than the entropic map which does learn the correct path. Thus, the

	Mean	Var
Prospect	0.026	0.019
Entropic Map	0.095	4.3e-3
Log	0.021	1e-4

TABLE II: Error mean and variance, as measured by the L_1 norm, between learned and true behavior for an true prospect agent learned by prospect, entropic, and logarithmic agents.

log value function may generalize better, since it has learned the policy in general, better than the entropic map. Further, we note that the log function better recreates the policy than the prospect value function. This is most likely due to the fact that the log value function approximates the prospect value function, but is globally Lipschitz. Thus, we have stronger guarantees of convergence which translate into better results.

B. A Passenger's View of Ride-Sharing

We derive a toy model based on pricing data collected from the Uber API and travel time data collected via Uber Movement[2]. The pricing data was collected at 3-minute intervals across 276 locations in Washington, D.C. (specified by Census blocks) from 2016 November 14–28[3].

From the passenger's view point, we model the ride-sharing MDP as follows. The action space is $A = \{\text{ride}, \text{wait}\}$ and the state space $X = \mathscr{X} \times \mathscr{T} \cup \{x_f\}$ where $\mathscr{X} = \{1.0, 1.4, 1.8, 2.2\}$ is the part of the state corresponding to the price multiplier, $\mathscr{T} = \{0, \ldots, T_f\}$ is the part of the state corresponding to the time index, and x_f is a terminal state representing the completed ride that occurs when a ride is taken. At time t, the state is notationally given by (x_t, t). The reward r_t is modeled as a random variable that depends on the current price as well as a random variable $Z(t)$ for travel time whose distribution is estimated from the Uber Movement travel time data. In particular, for any time $t < T_f$ the reward is given by

$$R(x_{t+1}, a_t, x_t) = \begin{cases} \bar{r}, & a_t = \text{wait} \\ \tilde{r}(t), & a_t = \text{ride} \end{cases} \tag{18}$$

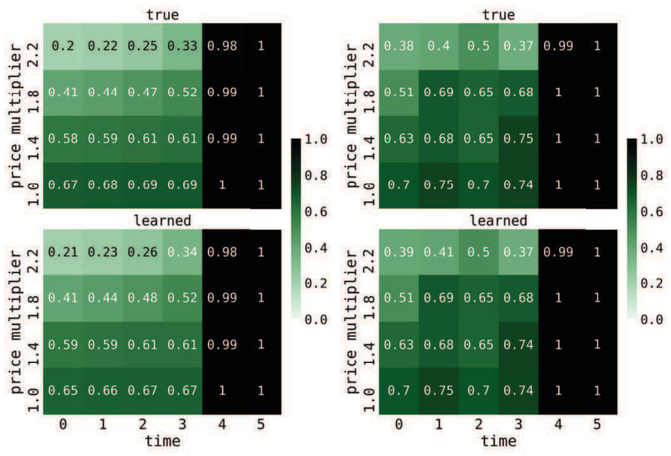
with $\bar{r} < 0$ a constant and $\tilde{r}(t) = S_t - x_t(p_{\text{base}} + p_{\text{mile}} D + p_{\text{min}} Z(t))$ where D is the distance in miles, S_t is a time dependent satisfaction (we selected it to linearly decrease in time from some initial satisfaction level), and p_{base}, p_{mile}, and p_{min} are the base, per mile, and per min prices, respectively. These prices are chosen based on Washington, DC Uber prices[4]. At the final time T_{final}, the agent is forced to take the ride if they have not selected to take a ride at a prior time. This reflects the fact that the agent presumably needs to get from their origin to their destination and the reward structure reflects the dissatisfaction the agent feels as a result of having to ultimately take the ride despite the potential desire to wait.

The transition probability kernel $P : X \times A \times X \to [0, 1]$ is estimated from the ride-sharing data. The travel-time data we

[2]Uber Movement: `https://movement.uber.com/cities`

[3]The data was originally collected by and has been made publicly available here: `https://github.com/comp-journalism/2016-03-wapo-uber`

[4]The base, per min, and per mile prices can be found here: `http://uberestimate.com/prices/Washington-DC/`



(a) risk-neutral, $\rho_+ = \rho_- = 1.0$ (b) risk-seeking in losses, $\rho_- = 0.5$

Fig. 3: Prospect Agents with $c_+ = c_- = 1.0$ and $\rho_+ = 1.0$. The plots show the probabilities of taking a ride in each state under the true and learned optimal policies. The more risk-seeking an agent is in losses, the more likely they are to take a ride.

have is available on an hourly basis. Hence, we use the 3 min price change data for each hour to derive a static transition matrix by empirically estimating the transition probabilities where we bin prices in the following way. For prices in $[1.0, 1.2)$, $x = 1.0$; for prices in $[1.2, 1.6)$, $x = 1.4$; for prices in $[1.6, 2.0)$, $x = 1.8$; otherwise $x = 2.2$. In the time periods we examined, the max price multiplier was 2.0. We selected the reference point to be $r_0 = 0$.

We examined several locations and hours which had different characteristics in terms of travel time and price statistics. However, the core risk-sensitive behaviors were the same. For the examples depicted in Fig. 3, we generated a ride-sharing MDP for Washington D.C. origin GPS$=(-77.027046, 38.926749)$ and destination GPS$=(-76.935773, 38.885964)$[5] at 5AM.

The transition matrix for the price multipliers is given by

$$P = \begin{bmatrix} 0.876 & 0.099 & 0.017 & 0.008 \\ 0.347 & 0.412 & 0.167 & 0.074 \\ 0.106 & 0.353 & 0.259 & 0.282 \\ 0.086 & 0.219 & 0.143 & 0.552 \end{bmatrix} \tag{19}$$

for each time. The travel time distribution is a standard normal distribution truncated to the upper and lower bounds specified by the Uber Movement data. Measured in seconds, we use location parameter 2371, scale parameter 100, and 1554 and 3619 as the upper and lower bound, respectively.

In Fig. 3, for a passenger (assumed to be an agent with a prospect value) that is risk-averse, risk-neutral, and risk-seeking in losses and risk neutral in gains, we show the state space as a grid with the probability of taking a ride under the true and learned optimal policy overlaid on each state. As the agent becomes more risk-seeking in losses, they are less likely to take the ride when in a high price state and more likely to take the ride when in a low price state. This is likely due to the fact that they are more willing to *risk it* in high price states by waiting for a lower price and in low price states by jumping on the low price opportunity.

[5]Note that these correspond to Uber Movement id's 197 and 113, respectively.

V. Discussion

We presented a new gradient based technique for learning risk-sensitive decision making models of humans amidst automation. We find that while there are a number of technical issues related to learning prospect theory based agents—namely, their value functions are not Lipschitz for parameter combinations that best capture human decision making—we are able to still numerically learn the policies of these agents. Moreover, we introduce a logarithm-based value function that well-approximates the convex-concave structure of the prospect theory values while satisfying the assumptions of our theorems. We demonstrated the algorithms performance for agents based on several types of behavioral models and do so on two test cases.

References

[1] A. Tversky and D. Kahneman, "Rational Choice and the Framing of Decisions," *J. Business*, vol. 59, no. 4, pp. S251–S278, 1986.
[2] ——, "Loss aversion in riskless choice: A reference-dependent model," *Quarterly J. Economics*, vol. 106, no. 4, pp. 1039–1061, 1991.
[3] Y. Shen, W. Stannat, and K. Obermayer, "Risk-Sensitive Markov Control Processes," *SIAM J. Control Optimization*, vol. 51, no. 5, pp. 3652–3672, 2013.
[4] S. I. Marcus, E. Fernández-Gaucherand, D. Hernández-Hernández, S. Coraluppi, and P. Fard, *Risk Sensitive Markov Decision Processes*. Birkhäuser Boston, 1997.
[5] P. L.A., C. Jie, M. Fu, S. Marcus, and C. Szepesvári, "Cumulative prospect theory meets reinforcement learning: Prediction and control," in *Proc. 33rd Intern. Conf. on Machine Learning*, vol. 48, 2016.
[6] Y. Shen, M. J. Tobia, and K. Obermayer, "Risk-sensitive reinforcement learning," *Neural Computation*, vol. 26, pp. 1298–1328, 2014.
[7] O. Mihatsch and R. Neuneier, "Risk-sensitive reinforcement learning," *Machine Learning*, vol. 49, no. 2, pp. 267–290, 2002.
[8] A. J. Nagengast, D. A. Braun, and D. M. Wolpert, "Risk-sensitive optimal feedback control accounts for sensorimotor behavior under uncertainty," *PLOS Computational Biology*, vol. 6, no. 7, pp. 1–15, 2010.
[9] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–291, 1979.
[10] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *J. Risk Uncertainty*, vol. 5, no. 4, pp. 297–323, Oct 1992.
[11] H. Simon, "Bounded rationality in social science: Today and tomorrow," *Mind & Society*, vol. 1, no. 1, pp. 25–39, Mar. 2000.
[12] A. Tversky and D. Kahneman, "The framing of decisions and the psychology of choice," *Science*, vol. 211, no. 4481, pp. 453–458, Jan. 1981.
[13] C. F. Camerer, "An experimental test of several generalized utility theories," *J. Risk and Uncertainty*, vol. 2, no. 1, pp. 61–104, 1989.
[14] H. Föllmer and A. Schied, "Convex measures of risk and trading constraints," *Finance and Stochastics*, vol. 6, no. 4, pp. 429–447, 2002.
[15] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.
[16] H. Föllmer and A. Schied, *Stochastic Finance: An Introduction in Discrete Time*. Walter de Gruyter, 2004.
[17] L. J. Ratliff, E. Mazumdar, and T. Fiez, "Risk-sensivtive inverse reinforcement learning via gradient methods," *arXiv*, 2017.