2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Risk-Sensitive RL Using Sampling-Based Expectation-Maximization

Erfaun Noorani[1], John S. Baras[1], and Karl H. Johansson[2]

Abstract— **There is a need for robust Reinforcement Learning (RL) algorithms that can cope with model misspecification, parameter uncertainty, disturbances, etc. Risk-sensitive methods offer an approach to developing robust RL algorithms by hedging against undesirable outcomes in a probabilistic manner. The Probabilistic Graphical Model (PGM) framework offers systematic exploration for risk-sensitive RL. In this paper, we bridge the Markov Decision Process (MDP) and the PGM frameworks. We exploit the equivalence of optimizing a certain risk-sensitive criterion in the MDP formalism with optimizing a log-likelihood objective in the PGM formalism. By utilizing this equivalence, we offer an approach for developing risk-sensitive algorithms by leveraging the PGM framework. We explore the Expectation-Maximization (EM) algorithm under the PGM formalism. We show that risk-sensitive policy gradient methods can be obtained by applying sampling-based approaches to the EM algorithm, e.g., Monte-Carlo EM, with the log-likelihood. We show that Monte-Carlo EM leads to a risk-sensitive Monte-Carlo policy gradient algorithm. Our simulations illustrate the risk-sensitive nature of the resulting algorithm.**

I. Introduction

Online learning for decision-making and control has become critical due to the increasing complexity of systems, which renders approaches based solely on models inadequate. Furthermore, changing environments, unknown environments, and incorrect task execution, also necessitate online learning. Reinforcement Learning (RL) offers a promising response to the need for online learning. RL studies the design and implementation of optimal policies using interactions with an (unknown or complex) environment [1]. Uncertainty in stochastic environments leads to the stochastic performance of policies. Risk-neutral RL algorithms aim to find a policy with the highest expected performance. Risk-sensitive RL algorithms consider a criterion beyond the expected performance, e.g., mean-variance [2] and exponential criterion [3].

Though risk-neutral RL algorithms provide an effective approach to data-driven control, it is well-known that risk-neutral methods are brittle, i.e., non-robust, both in the context of optimization and model-based control and more recently in the context of RL and data-driven control [4]. The non-robust performance of risk-neutral algorithms has led to a surge of interest in robust RL algorithms that can cope with model misspecification, parameter uncertainty, disturbances, etc. The robust properties of risk-sensitive algorithms and their practical advantage over robust (worst-case) control algorithms [5] make risk-sensitive approaches particularly suitable for real-world applications. For a recent research monograph on risk-sensitive RL, see [6].

The Markov Decision Process (MDP) [7] has become conventional for modeling RL and the de facto language in developing both risk-neutral and risk-sensitive RL algorithms. The tools and techniques for handling uncertainty within the MDP framework have been instrumental in the success of RL. The Probabilistic Graphical Model (PGM) provides an alternative approach to modeling RL problems. PGMs offer a rich set of tools and techniques for inference and learning, e.g., the Expectation-Maximization (EM) algorithm, its variants, and alternatives. For a tutorial on EM algorithms, see [8], [9]. An elegant analysis of the EM algorithm and its convergence can be found in [10], where its fundamental nature as gradient descent in an appropriate space is established. A recent tutorial on modeling RL using PGMs is given in [11].

The use of EM-inspired algorithms in RL has been previously explored, e.g., see [12]. The survey in [13] presents some results on the EM algorithms for policy search. These results make no connection with risk-sensitive RL. Casting RL as a PGM has allowed for establishing an equivalence between the optimization of the exponential criterion in the MDP framework and a certain log-likelihood in the PGM framework [14]. These results focus on establishing the equivalence, and it does not offer any algorithm or discussion on the implications of this equivalence. This motivates us to ask:

"How can we capitalize on this equivalence to contribute to algorithm design for risk-sensitive RL?"

Contributions. We discuss our contributions in detail here:

- We build upon our results in [14] and establish the equivalence between the exponential criteria with a negative risk parameter (risk-averse behavior) under the MDP framework and a certain log-likelihood under the PGM framework. The result in [14] establishes such equivalence for the exponential criteria with a positive risk parameter (risk-seeking behavior). We summarize these results in Proposition 1.
- By leveraging this equivalence, we employ sampling-based EM algorithms and stochastic gradient schemes to design a risk-sensitive policy search algorithm. The risk-sensitive nature of our algorithm stems from the log-likelihood objective function that was derived from the mathematical analysis of the risk-sensitive exponential criteria under the PGM model. By doing this, we arrive at the update rule (8). We find the risk-sensitive algorithm recently proposed in [15] is naturally recovered out of the PGM framework.

[1]E. Noorani and J. Baras are with the Department of Electrical and Computer Engineering and the Institute for Systems Research (ISR) at the University of Maryland, College Park, MD, USA. {enoorani,baras}@umd.edu.

[2]Karl H. Johansson is with the Automatic Control Lab, Royal Institute of Technology, Stockholm, Sweden. {kallej}@kth.se.

Research partially supported by ONR grant N00014-17-1-2622, by a grant from the Army Research Lab, and by the Clark Foundation.

This shows that the algorithm in [15] is a Monte-Carlo EM algorithm. This is significant in terms of algorithmic analysis since the strong guarantees associated with EM algorithms such as convergence and its weaknesses such as slow convergence under certain conditions [16], can be attributed to this update rule.

- We explain the robust performance of the update rule (8) by showing that this update rule is an EM algorithm for solving the risk-sensitive RL with exponential criteria. Our analysis also shows that the postulated choice of exponential function for transforming the trajectory rewards into probability distributions in [13], [17], leads to the optimization of the exponential criteria and suggests that different choices for the transforming functions lead to the optimization of different risk-measures.

Our contributions bridge the MDP and PGM, and provide a systematic approach to the further development of risk-sensitive RL using PGMs. Our analysis suggests that the EM algorithm, its variants, and alternatives can be used to design a host of risk-sensitive RL algorithms. An additional advantage offered by the PGM framework is the hierarchical decomposition of control and decision-making problems, "divide and conquer" approaches and modularity, which are often necessary for the increasing complexity of systems, and problems, encountered. We point out an instance of "divide and conquer" in Section VII.

II. Reinforcement Learning

MDPs are the conventional framework for modeling RL [7]. Most risk-neutral RL algorithms and, to the best of our knowledge, all risk-sensitive RL algorithms are developed using the MDP framework. PGM is an alternative framework for modeling RL problems. We model the RL problem, first, using MDP, and then, using PGM. In doing so, we present some background materials and set our notation.

A. MDP

An MDP is described by the tuple $M = (\mathcal{S}, \mathcal{A}, p_1, P, r)$, where $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space, p_1 is the initial state distribution, P is the transition kernel, and r is the immediate reward function.

Under the MDP framework, the environment is modeled as a discrete-time stochastic process. The process starts in an initial state s_1 given by the initial probability distribution p_1. For a given action a from the set of admissible actions (action space), the environment transitions from the current state s to a successor state s' with a probability given by $p(s'|s,a)$. A scalar reward $r(s,a)$ is given when the action is executed. Over time, the system goes through a sequence of states and actions. This sequence is called a trajectory and is denoted by τ, i.e., $\tau := (s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T)$, where T is the system time horizon.

The behavior of the RL agent is characterized by its policy, which could be deterministic or randomized. A (randomized) policy $\pi(\cdot|s)$ prescribes the probability of taking an action when in a given state. The policy can be represented by a differentiable parameterized function of the state $\pi_\theta(\cdot|s)$ where $\theta \in \mathbb{R}^d$ is a vector of d parameters.

Under a given policy, each trajectory τ happens with a probability induced by the agents' policy and the system transition probabilities, and is given by

$$p_\theta(\tau) = p_1 \prod_{t=1}^{T} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t). \tag{1}$$

The agent aims to optimize some desired measure of performance. In classical (risk-neutral) RL, the desired performance criterion is the expected performance. A common example of a risk-neutral objective in the RL literature is the expected (discounted) cumulative reward,

$$J(\theta) := \mathbb{E}\Big[R\Big], \tag{2}$$

where the expectation is taken under policy's trajectory distribution, i.e., $s_1 \sim p_1$, $a_t \sim \pi_\theta(\cdot|s_t)$ and $s_{t+1} \sim p(\cdot|s_t, a_t)$, and the discounted cumulative reward $R := \sum_{t=1}^{T} \gamma^{t-1} r_t$, where $r_t := r(s_t, a_t)$, is the discounted sum of all per-stage rewards during an episode. The discount factor is $\gamma \in (0, 1]$.

Risk-sensitive algorithms use a criterion beyond the expectation. We will discuss the risk-sensitive RL in more detail in Section III. It is important to emphasize that the per-stage reward and the performance criterion should be designed to reflect the task at hand.

B. PGM

A PGM consists of an acyclic directed graph $G=(V, E)$ and a set of properties that determine a family of probability distributions. The sets V and E denote the set of nodes and edges of the graph, respectively. Each node represents a random variable. The edges represent conditional dependence assumptions, e.g., an edge from the random variable s_1 to s_2 indicates the dependence of the random variable s_2 on s_1 (see Fig. 1). For nodes a and s in the set V, node a is a parent of node s if and only if there exists an edge from node a to node s, e.g., a_t is a parent of s_{t+1} for any t in Fig. 1. The dependence between a random variable and its parents is typically defined as a conditional distribution of the random variable represented by the node, e.g., factors of the form $p(s_{t+1}|s_t, a_t)$ for the node s_{t+1} in Fig. 1.

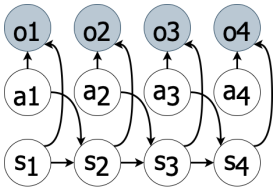

Fig. 1. RL in the PGM Framework.

PGM can be used as an alternative framework for modeling RL problems, resulting in a PGM with factors of the form $p(s_{t+1}|s_t, a_t)$. Any feedback system can be unrolled in time and be represented by a graphical model, e.g., a Bayesian Network. This models the relationship between state s_t, action a_t and successor-state s_{t+1}. Then by introducing a fictitious binary optimality variable denoted by o_t at each time

step, one can model the notion of reward into the graphical model; see Fig. 1 for a pictorial representation. Note that the system trajectory, the sequence of states s_t and actions a_t, are observable variables and represented using white nodes. The optimality variables are unobserved variables and are represented by gray nodes. By conditioning on the optimality variables to be true, one can infer the most probable policy. See [11] for a recent tutorial on modeling RL using PGM.

The optimality variable is equal to one, o_t=1, if the optimal action is taken at time step t and is equal to zero, o_t=0, if a non-optimal action is taken at time step t (hence the name optimality variable). For brevity, we use o_t and o'_t to denote o_t=1 and o_t=0, respectively. We use $O_{1:T}$ to denote the event that the optimal action was taken at each time step during an episode, i.e., $O_{1:T}$=$(o_1,\cdots,o_T)$, and $O'_{1:T}$ to denote the event that the optimal action was not taken at all time steps during an episode, i.e., $O'_{1:T}$=$(o'_1,\cdots,o'_T)$. From (1), it follows that the joint probability of observing a trajectory and being optimal at all time steps $p_\theta(O_{1:T},\tau)$ is,

$$p_\theta(O_{1:T},\tau) = p_1 \prod_{t=1}^{T} p(a_t|s_t)p(s_{t+1}|s_t,a_t)p_\theta(o_t|s_t,a_t) \quad (3)$$

and the joint probability of observing a trajectory and not being optimal at all time steps is given by

$$p_\theta(O'_{1:T},\tau) = p_1 \prod_{t=1}^{T} p(a_t|s_t)p(s_{t+1}|s_t,a_t)p_\theta(o'_t|s_t,a_t) \quad (4)$$

where the action prior is denoted by $p(a_t|s_t)$. We assume that the action prior $p(a_t|s_t)$ is a constant corresponding to a uniform distribution over the action space. This assumption does not introduce any loss of generality, because any non-uniform prior $p(a_t|s_t)$ can be incorporated instead into $p_\theta(o_t|s_t,a_t)$ (resp. $p_\theta(o'_t|s_t,a_t)$) via the reward function, as we shall see. The choice of the probability distribution of the optimality variable conditioned on the state-action pair $p_\theta(o_t|s_t,a_t) = p(o_t = 1|a_t,s_t)$ and $p_\theta(o'_t|s_t,a_t) = p(o_t = 0|a_t,s_t)$ defines the meaning of optimality and therefore the objective function of the agent.

III. Risk-Sensitive RL

In this section, we briefly introduce the risk-sensitive exponential criterion (under the MDP framework) and state its connection with a certain log-likelihood objective under the PGM framework. This connection enables the development of risk-sensitive RL algorithms under the PGM framework.

Risk-sensitive RL algorithms use performance criteria that consider some notion of risk, e.g., higher moments of return. The exponential criterion is a particular example of such criteria and is given by

$$J_\beta(\theta) := \mathbb{E}\Big[\beta e^{\beta R}\Big], \quad (5)$$

where the expectation is taken under policy's trajectory distribution and $\beta \in \mathbb{R}$ is a real-value constant design parameter that controls the agent's risk-attitude. The agent shows risk-averse behavior for a negative risk-parameter β<0 and risk-seeking behavior for a positive risk-parameter β>0 [3]. A Taylor expansion of the exponential function shows that the exponential criterion is a weighted infinite sum given by

$$\mathbb{E}\Big[\beta e^{\beta R}\Big] = \beta + \beta^2 \mathbb{E}\Big[R\Big] + \frac{\beta^3}{2}\mathbb{E}\Big[R^2\Big] + \dots \quad .$$

The risk-sensitive optimal policy converges to the risk-neutral optimal policy in limit when the risk parameter β approaches zero [3]. This can be discerned by the Taylor expansion of the exponential function and the fact that the optimal solution doesn't change when the objective function shifted by a constant (e.g. β) or when scaled by a positive scalar (e.g. a factor of $1/\beta^2$).

The results in [14] give a probabilistic interpretation of maximizing the exponential criterion (5) by casting the risk-sensitive RL problem into the PGM framework, and establish that, under a bounded reward structure $r\in[r_{min}, r_{max}]$, the maximization of the exponential criterion is equivalent to maximizing the probability of taking an optimal action at all time steps during an episode for a factored form of

$$p_\theta(o_t \mid s_t, a_t) := \pi(a_t \mid s_t;\theta)e^{\beta r_t}.$$

That is, the specific choice of $p_\theta(o_t \mid s_t, a_t) := \pi(a_t|s_t;\theta)e^{\beta r_t}$ maps back to a specific objective function in the MDP framework, namely, the exponential criterion with a positive risk parameter $\beta > 0$ (risk-seeking behavior). [14] does not leverage this equivalence and offers no algorithm. To leverage this equivalence and design a risk-sensitive RL algorithm, we first extend the results in [14] to characterize both positive (risk-seeking behavior) and negative (risk-averse behavior) risk parameters and formally state this connection in Proposition 1. We then use this equivalence to leverage EM algorithms for developing risk-sensitive RL in the proceeding section.

Proposition 1: Under the assumption of bounded reward structure, for the choice of $p(o_t|s_t,a_t) = \pi_\theta(a_t|s_t)e^{\beta r_t}$ with the temperature parameter $1/\beta > 0$, we have

$$\operatorname*{argmax}_\theta J_\beta(\theta) = \operatorname*{argmax}_\theta \log p_\theta(O_{1:T}), \quad \forall \beta > 0$$

and for the choice of $p(o'_t|s_t,a_t) = \pi_\theta(a_t|s_t)e^{\beta r_t}$ with the temperature parameter $1/\beta < 0$, we have

$$\operatorname*{argmax}_\theta J_\beta(\theta) = \operatorname*{argmin}_\theta \log p_\theta(O'_{1:T}), \quad \forall \beta < 0.$$

Proof: For the β<0 case, from the premise of the theorem, we have $p(o'_t \mid s_t,a_t)$=$\pi(a_t \mid s_t;\theta)e^{\beta r_t}$ where β is a negative constant. Also, recall that $p(O'_{1:T},\tau) = p_1 \prod_{t=1}^{T} p(a_t \mid s_t)p(s_{t+1} \mid s_t,a_t)p(o'_t|s_t,a_t)$ (cf. Eq. (4)), and $p_\theta(\tau)$=$p_1 \prod_{t=1}^{T} \pi(a_t|s_t;\theta)p(s_{t+1}|s_t,a_t)$ (cf. Eq. (1)). Thus, using the property of exponential, we have

$$\begin{aligned} p(O'_{1:T}) = \int_\tau p(O'_{1:T},\tau)d\tau &\propto \int_\tau p_\theta(\tau)e^{\beta R(\tau)}d\tau \\ &\propto -\mathbb{E}_{\tau\sim p_\theta}\Big[\beta e^{\beta R(\tau)}\Big] \end{aligned}$$

The equality follows from the definition of the marginal distribution. The first proportionality follows from Eq. (4) and the constant action priors. The last line follows immediately from the definition of expectation and the negativity of β.

The $\beta>0$ case has been proven in [14] and follows a similar logic as shown here. ■

Remark 1: Proposition 1 suggests that maximizing the joint probability of taking an optimal action at all time steps during an episode, i.e.,

$$\underset{\theta}{\operatorname{argmax}} \log p_\theta(O_{1:T}), \tag{6}$$

results in risk-seeking behavior, and minimizing the joint probability of not taking an optimal action at all time steps during an episode, i.e.,

$$\underset{\theta}{\operatorname{argmin}} \log p_\theta(O'_{1:T}), \tag{7}$$

results in risk-averse behavior. This offers a probabilistic perspective on risk-sensitive RL and justifies the choice of exponential criterion in RL.

The equivalence stated in Proposition 1 suggests an avenue for developing risk-sensitive RL algorithms by leveraging numerous existing methods for optimizing log probabilities. A wealth of tools and algorithms for inference and learning on graphical models, e.g. the EM algorithm, have been developed over the years [8], [9]. These tools can be used to (approximately) solve the risk-sensitive RL problem. The objective functions in (6) and (7) are derived from the mathematical analysis of the risk-sensitive exponential criterion under the PGM framework and are different from the objective functions used in methods described in [13].

IV. Monte-Carlo EM and Risk-Sensitive RL

We focus on the EM algorithm and its sampling-based variant, Monte-Carlo EM. We will explore other EM variants and alternatives for developing risk-sensitive RL algorithms in our future work. We formally state our result in Theorem 1. The proof of Theorem 1 is presented in Section V.

Theorem 1: Let $R_t := \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}$ denote the reward-to-go at time t and let α be the step size. Then, the iteration

$$\theta^{k+1} = \theta^k + \alpha e^{\beta R_t} \nabla_\theta \log \pi_\theta(a_t|s_t) \tag{8}$$

constitutes a Monte-Carlo EM algorithm designed for solving risk-sensitive exponential criteria (5).

Proof: The proof uses the sampling-based Monte-Carlo EM and gradient descent algorithms. See Section V for a step-by-step proof. ■

Remark 2: The update rule in Theorem 1 is the risk-sensitive policy gradient algorithm proposed in [15] that has been derived using the MDP framework. This suggests that risk-sensitive policy gradient algorithms can be thought of as sample- and gradient-based EM algorithms.

V. Proof of Theorem 1

Here, we provide a step-by-step proof of the theorem. The proof is the application of the EM algorithm to the RL problem under the PGM framework. We first state a lemma. The lemma is later used to construct a lower bound on the posterior distribution for the EM algorithm. We write the proof for the risk-seeking $\beta > 0$. The proof for the risk-averse $\beta < 0$ is analogous by substitution of $O'_{1:T}$ for $O_{1:T}$ and noting the negativity of β.

Lemma 1: [14] The log-likelihood in Eq. (6) can be decomposed as the sum of the Evidence lower-bound on the probability of being optimal at all time steps and a KL-divergence term, given by

$$\log p_\theta(O_{1:T}) = L(\theta) + D(\theta)$$

where

$$L(\theta) := \mathbb{E}_{\tau\sim p_\theta}\Big[\log p(O_{1:T}|\tau)\Big]$$

$$D(\theta) := D_{KL}\Big(p_\theta(\tau)\|p_\theta(\tau|O_{1:T})\Big)$$

$D_{KL}(Q,P)$ is the KL-divergence between the probability distributions Q and P.

Remark 3: It should be noted that the optimality of a trajectory is not dependent on the policy, that is to say, $p(O_{1:T}|\tau)$ is independent of the policy, and thus, does not depend on the policy parameters θ.

Remark 4: It is immediate from Lemma 1 and the non-negativity of KL divergence that $L(\theta)$ is a lower bound on the log probability.

We now present a step-by-step derivation and discussion of the theorem. Recall that in the EM algorithm, the E-step constructs a tractable lower bound $B(\theta,\theta_t)$ and the M-step maximizes the constructed bound. The E-step constructs a posterior distribution lower bound, see Lemma 1. The E-step updates the trajectory distribution $p_\theta(\tau)$ by minimizing the KL divergence term $D_{KL}(p_\theta(\tau)\|p_{\theta^k}(\tau|O_{1:T}))$, i.e.,

$$\theta^*_e := \underset{\theta}{\operatorname{argmin}} D_{KL}\Big(p_\theta(\tau)\|p_{\theta^k}(\tau|O_{1:T})\Big).$$

The KL divergence is non-negative and is minimized when the parameters are chosen such that $p_{\theta^*_e}(\tau) = p_{\theta^k}(\tau|O_{1:T})$. This lower bound is tight after each E-step. Thus, the M-step maximizes the constructed lower bound, i.e.,

$$\theta^{k+1} = \underset{\theta}{\operatorname{argmax}}\, B(\theta,\theta^k)$$

where B is a lower bound and is given by

$$B(\theta,\theta^k) := \mathbb{E}_{p_{\theta^k}(\tau|O_{1:T})}\Big[\log p_\theta(O_{1:T},\tau)\Big].$$

By noting that $p(O_{1:T}|\tau)$ is independent of the policy parameters θ, we have

$$p_\theta(O_{1:T},\tau) = p(O_{1:T}|\tau)p_\theta(\tau)$$

and by Bayes' rule, we have

$$p_{\theta^k}(\tau|O_{1:T}) = \frac{p(O_{1:T}|\tau)p_{\theta^k}(\tau)}{p_{\theta^k}(O_{1:T})}.$$

Thus,

$$\begin{aligned} B(\theta,\theta^k) &= \mathbb{E}_{\tau\sim p_{\theta^k}}\Big[\frac{p(O_{1:T}|\tau)}{p_{\theta^k}(O_{1:T})}\log p_\theta(O_{1:T},\tau)\Big] \\ &= \mathbb{E}_{\tau\sim p_{\theta^k}}\Big[\frac{p(O_{1:T}|\tau)}{p_{\theta^k}(O_{1:T})}\log p_\theta(\tau)\Big] \\ &\quad + \mathbb{E}_{\tau\sim p_{\theta^k}}\Big[\frac{p(O_{1:T}|\tau)}{p_{\theta^k}(O_{1:T})}\log p(O_{1:T}|\tau)\Big]. \end{aligned}$$

The second term is not a function of the decision variable θ, thus, we have

$$\theta^{k+1} = \operatorname*{argmax}_{\theta} \mathbb{E}_{\tau\sim p_{\theta^k}}\left[\frac{p(O_{1:T}|\tau)}{p_{\theta^k}(O_{1:T})}\log p_\theta(\tau)\right].$$

By noting $p_{\theta^k}(O_{1:T})$ is constant with respect to τ and not a function of θ (it is a function of θ^k), we have

$$\theta^{k+1} = \operatorname*{argmax}_{\theta} \mathbb{E}_{\tau\sim p_{\theta^k}}\left[p(O_{1:T}|\tau)\log p_\theta(\tau)\right].$$

Now recall that

$$p(O_{1:T}|\tau) = \prod_{t=1}^{T} p(o_t|s_t,a_t) = \prod_{t=1}^{T} e^{\beta r_t} = e^{\beta R}.$$

Thus,

$$\theta^{k+1} = \operatorname*{argmax}_{\theta} \mathbb{E}_{\tau\sim p_{\theta^k}}\left[e^{\beta R}\log p_\theta(\tau)\right].$$

By noting that $\log p_\theta(\tau) \overset{c}{=} \sum_{t=1}^{T}\log\pi_\theta(a_t|s_t)$ up to a constant term (constant with respect to the policy parameters), we arrive at the following update rule:

$$\theta^{k+1} = \operatorname*{argmax}_{\theta} \mathbb{E}_{\tau\sim p_{\theta^k}}\left[e^{\beta R}\sum_{t=1}^{T}\log\pi_\theta(a_t|s_t)\right]. \tag{9}$$

Remark 5: The update rule (9) is used in [17] and [13], but the connection with risk-sensitive RL has not been made.

Remark 6 (MM Principle): It is well known that EM algorithms can be thought of as an instance of Majorization-Minimization (MM) algorithms. The bound $B(\theta,\theta^k)$ minorizes the log-likelihood $\log p_\theta(O_{1:T})$, that is, it is tangent to the log-likelihood at a given policy parameter θ^k and it is dominated by it at all points, i.e.,

$$B(\theta^k,\theta^k) = \log p_{\theta^k}(O_{1:T}), \quad B(\theta,\theta^k) \le \log p_\theta(O_{1:T})$$

for all $\theta\in\mathbb{R}^d$. This suggests one could use alternative MM-type algorithms to develop risk-sensitive RL agents.

The iteration (9) is an EM algorithm to solve the risk-sensitive exponential criteria (5). To adapt the update rule of (9) to use the interactions with the environment, one can use sampling-based EM algorithms, such as Monte-Carlo EM. This optimization may be attempted by a variety of methods. One particularly easy-to-implement approach is using a first-order (gradient-based) method. The expectation in iteration (9) is with respect p_{θ^k} and independent of θ. This simplifies the gradient computation. The gradient ascent iteration can be written as

$$\theta_{t+1}^{k+1} = \theta_t^{k+1} + \alpha\mathbb{E}_{\tau\sim p_{\theta^k}}\left[e^{\beta R}\sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t)\right].$$

where $\theta_0^{k+1} = \theta^k$. The step size α is called the learning rate. Special consideration needs to be given to the step size α as its choice affects the learning process, as is the case for any stochastic approximation scheme [18]. This iteration could be further modified and written in terms of reward-to-go

$$\theta_{t+1}^{k+1} = \theta_t^{k+1} + \alpha\mathbb{E}_{\tau\sim p_{\theta^k}}\left[e^{\beta R_t}\sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t)\right].$$

To see this, from the iteration above and the property of exponential functions, we have

$$\begin{aligned}\theta^{k+1} &= \theta^k + \alpha\mathbb{E}_{\tau\sim p_{\theta^k}}\left[e^{\beta R_t^-}e^{\beta R_t}\sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t)\right]\\ &= \theta^k + \alpha\Big(\mathbb{E}_{\tau\sim p_{\theta^k}}\left[e^{\beta R_t^-}\right]\\ &\qquad \mathbb{E}_{\tau\sim p_{\theta^k}}\left[e^{\beta R_t}\sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t)\right]\Big)\end{aligned}$$

where R_t is the reward-to-go and R_t^- is the accumulated reward up to time t, i.e., $R{=}R_t{+}R_t^-$. The last line is due to the temporal structure that leads to the decorrelation between the random variables. Then one can attempt to estimate this expectation with temporal-difference or Monte Carlo methods. Since both the EM and the gradient-based methods are iterative methods, the computation can be reduced by taking a limited number of gradient steps at each iteration of the overall algorithm.

Using the full trajectory samples for estimating the reward-to-go R_t leads to a risk-sensitive *Monte-Carlo* algorithm– such an algorithm has been developed in [15] using the MDP framework. A stochastic approximation approach results in the update rule given by

$$\theta_{t+1} = \theta_t + \alpha e^{\beta R_t}\nabla_\theta\log\pi_\theta(a_t|s_t)$$

where an entire trajectory is generated and then samples of the reward-to-go R_t are used to make an update. This algorithm, as with all Monte Carlo methods, suffers from high variance. Several variance reduction techniques have been introduced for risk-neutral Monte-Carlo methods, yet there is currently a gap in the application of such techniques to our risk-sensitive method. It's worth highlighting that our method already exhibits superior variance characteristics when compared to risk-neutral approaches. As we advance our research agenda, we anticipate embracing sample-based methodologies and batch learning as promising avenues for further variance reduction.

Note that, by the equivalence stated in Proposition 1, the log-likelihood optimization of (6) when $\beta>0$ and of (7) when $\beta<0$ is equivalent to the optimization of the risk-sensitive exponential criteria (5). This concludes the proof.

VI. Numerical Example

The fact that the update rule in Theorem 1 is indeed risk-sensitive, in the sense that it considers the tail of the distribution, can be understood from the relation between the log-likelihood and the exponential criteria. To illustrate the risk-sensitivity of the update rule of Theorem 1, we compare the performance of the risk-neutral Monte Carlo policy gradient algorithm, REINFORCE [19], with the risk-sensitive update rule of Theorem 1 on the well-known RL benchmark of Cart-pole (inverted pendulum).

The agent's goal is to balance a pole mounted on a moving cart with an unactuated joint by pushing the cart to the left and right. The state variable consists of the position and velocity of the cart and the angle with the vertical and angular

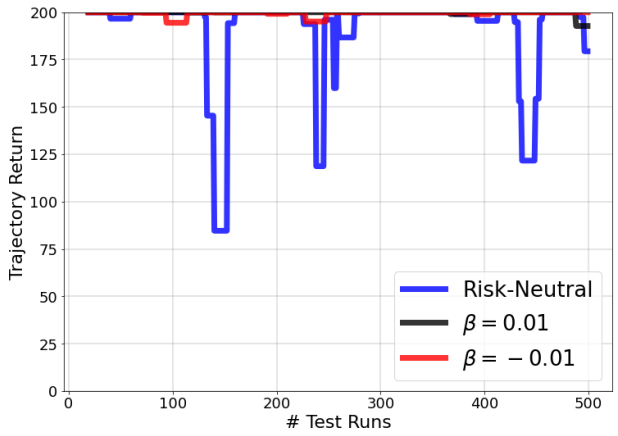

Fig. 2. The behavior of a trained risk-neutral agent, risk-seeking agent $\beta = 0.01$ and risk-averse agent $\beta = -0.01$ during testing in the cart-pole problem. The 0.05-quantile of the trajectory returns, calculated using a moving window of length 20, for 500 independent test runs are plotted.

velocity of the pole. The admissible actions are a "left" or "right" force of fixed magnitude. A reward of +1 is given for each time step that the pole is kept upright. The agent is successful if it manages to keep the pole upright for 200 time steps and it fails when the pole deviates from the vertical for more than 12 degrees or when the position of the cart is more than 2.4 meters away from the starting point.

The policy is parameterized by a neural network with one hidden layer of 16 neurons and a 'ReLU' activation function. The learning rate is set to α=0.01 and the discount factor to γ=0.99. The optimizer 'Adam' is used with decaying step sizes. We train the agent for 1000 episodes and run an additional 500 test runs.

Both risk-neutral and risk-sensitive methods achieve policies with an average training reward near 200. However, the advantage of risk-sensitive algorithms becomes apparent when examining the tail of the trajectory return distribution during test runs. For instance, consider selecting a point on the right tail of the return distribution. Let's say 150 time steps. Out of 500 test runs, the risk-neutral algorithm failed to balance the pole for more than 150 steps in 10 runs, while the risk-averse and risk-seeking algorithms failed to reach the 150-step threshold in only 1 and 2 runs, respectively. To better illustrate this point, we use the σ-quantile of the trajectory return defined as: $\inf\{x \in \mathbb{R} : P(R \leq x) > \sigma\}$.

Fig.2 shows the plots of 0.05-quantile of returns. The figure illustrates that the risk-sensitive algorithm ($\beta = \{+0.01, -0.01\}$) achieves higher σ-quantile (for $\sigma = 0.05$) than the risk-neutral algorithm. The quantile in the figure is calculated using a moving window of length 20 over the returns of the test runs. We observe that the risk-sensitive approach has a noticeable effect on the tail of the distribution.

VII. Conclusion

We have shown that risk-sensitive algorithms can be derived using the PGM formalism of RL. In this paper, we explore the utility of EM algorithms for such problems. EM algorithms have well-known advantages and disadvantages. For example, it is known that EM algorithms converge to the local optimum and have no guaranteed convergence rate. However, their simple and intuitive mechanism is the appeal of the EM algorithm . The separation of the E-Step and M-Step in the algorithm results in straightforward and interpretable procedures. For example, in the E-step, the expectation is computed with respect to the current parameters, rendering it constant during the subsequent M-step. This property greatly simplifies the mathematical aspects of the algorithm through a "divide and conquer" approach. In our future work, we will explore sampling-based EM algorithms that rely on step-wise samples. These algorithms may lead to RL algorithms with per-sample updates. We will also further explore the variants and alternatives to EM algorithms for developing risk-sensitive RL using the PGM modeling framework. Our main objective here was to provide a framework and a systematic approach to leverage PGM formalism for risk-sensitive RL, and also illustrate the approach and results in simple examples.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2018.

[2] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[3] R. A. Howard and J. E. Matheson, "Risk-Sensitive Markov Decision Processes," *Management Science*, vol. 18, no. 7, pp. 356–369, 1972.

[4] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete Problems in AI Safety," *arXiv preprint arXiv:1606.06565*, 2016.

[5] A. Nilim and L. El Ghaoui, "Robust Control of Markov Decision Processes with Uncertain Transition Matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[6] L. Prashanth, M. C. Fu *et al.*, "Risk-Sensitive Reinforcement Learning via Policy Gradient Search," *Foundations and Trends® in Machine Learning*, vol. 15, no. 5, pp. 537–693, 2022.

[7] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[8] R. M. Neal and G. E. Hinton, "A View of The EM Algorithm that Justifies Incremental, Sparse, and Other variants," in *Learning in graphical models*. Springer, 1998, pp. 355–368.

[9] F. Dellaert, "The Expectation Maximization Algorithm," Georgia Institute of Technology, Tech. Rep., 2002.

[10] L. Xu and M. I. Jordan, "On Convergence Properties of the EM Algorithm for Gaussian Mixtures," *Neural computation*, vol. 8, no. 1, pp. 129–151, 1996.

[11] S. Levine, "Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review," *arXiv:1805.00909*, 2018.

[12] J. Kober and J. Peters, "Policy Search for Motor Primitives in Robotics," *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[13] M. P. Deisenroth, G. Neumann, J. Peters *et al.*, "A Survey on Policy Search for Robotics," *Foundations and trends in Robotics*, vol. 2, no. 1-2, pp. 388–403, 2013.

[14] E. Noorani and J. S. Baras, "A Probabilistic Perspective on Risk-sensitive Reinforcement Learning," in *2022 American Control Conference (ACC)*, 2022, pp. 2697–2702.

[15] ——, "Risk-sensitive REINFORCE: A Monte Carlo Policy Gradient Algorithm for Exponential Performance Criteria," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 1522–1527.

[16] R. Salakhutdinov, S. Roweis, and Z. Ghahramani, "Expectation-conjugate gradient: An alternative to EM," *IEEE Signal Processing Letters*, vol. 11, no. 7, 2004.

[17] Y. Song and D. Scaramuzza, "Policy Search for Model Predictive Control With Application to Agile Drone Flight," *IEEE Transactions on Robotics*, 2022.

[18] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Springer, 2009, vol. 48.

[19] R. J. Williams and J. Peng, "Function Optimization using Connectionist Reinforcement Learning Algorithms," *Connection Science*, vol. 3, no. 3, pp. 241–268, 1991.