Being Optimistic to Be Conservative: Quickly Learning a CVaR Policy

Ramtin Keramati[1], Alex Tamkin,[2] Christoph Dann[3], Emma Brunskill[2]

[1]Institute of Computational and Mathematical Engineering (ICME), Stanford University, California, USA
[2]Department of Computer Science, Stanford University, California, USA
[3] Machine Learning Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
{keramati,atamkin,ebrun}@cs.stanford.edu, cdann@cdann.net

Abstract

While maximizing expected return is the goal in most reinforcement learning approaches, risk-sensitive objectives such as conditional value at risk (CVaR) are more suitable for many high-stakes applications. However, relatively little is known about how to explore to quickly learn policies with good CVaR. In this paper, we present the first algorithm for sample-efficient learning of CVaR-optimal policies in Markov decision processes based on the optimism in the face of uncertainty principle. This method relies on a novel optimistic version of the distributional Bellman operator that moves probability mass from the lower to the upper tail of the return distribution. We prove asymptotic convergence and optimism of this operator for the tabular policy evaluation case. We further demonstrate that our algorithm finds CVaR-optimal policies substantially faster than existing baselines in several simulated environments with discrete and continuous state spaces.

Introduction

A key goal in reinforcement learning (RL) is to quickly learn to make good decisions by interacting with an environment. In most cases the quality of the decision policy is evaluated with respect to its expected (discounted) sum of rewards. However, in many interesting cases, it is important to consider the full distributions over the potential sum of rewards, and the desired objective may be a risk-sensitive measure of this distribution. For example, a patient undergoing a surgery for a knee replacement will (hopefully) only experience that procedure once or twice, and may well be interested in the distribution of potential results for a single procedure, rather than what may happen on average if he or she were to undertake that procedure hundreds of time. Finance and (machine) control are other cases where interest in risk-sensitive outcomes are common.

A popular risk-sensitive measure of a distribution of outcomes is the Conditional Value at Risk (CVaR) (Artzner et al. 1999). Intuitively, CVaR is the expected reward in the worst α-fraction of outcomes, and has seen extensive use in financial portfolio optimization (Zhu and Fukushima

2009), often under the name "expected shortfall". While there has been recent interest in the RL community in learning to converge or identify good CVaR decision policies in Markov decision processes (Chow and Ghavamzadeh 2014; Chow et al. 2015; Tamar, Glassner, and Mannor 2015; Dabney et al. 2018a), interestingly we are unaware of prior work focused on how to quickly learn such CVaR MDP policies, even though sample efficient RL for maximizing expected outcomes is a deep and well studied theoretical (Jaksch, Ortner, and Auer 2010; Dann et al. 2018) and empirical (Bellemare et al. 2016) topic. Sample efficient exploration seems of equal or even more importance in the case when the goal is risk-averse outcomes.

In this paper we work towards sample efficient reinforcement learning algorithms that can quickly identify a policy with an optimal CVaR. Our focus is in minimizing the amount of experience needed to find such a policy, similar in spirit to probably approximately correct RL methods for expected reward. Note that this is different than another important topic in risk-sensitive RL, which focuses on safe exploration: algorithms that focus on avoiding any potentially very poor outcomes during learning. These typically rely on local smoothness assumptions and do not typically focus on sample efficiency (Berkenkamp et al. 2017; Koller et al. 2018); an interesting question for future work is whether one can do both safe and efficient learning of a CVaR policy. Our work is suitable for the many settings where some outcomes are undesirable but not catastrophic.

Our approach is inspired by the popular and effective principle of optimism in the face of uncertainty (OFU) in sample efficient RL for maximizing expected outcomes (Strehl and Littman 2008; Brafman and Tennenholtz 2002). Such work typically works by considering uncertainty over the MDP model parameters or state-action value function, and constructing an optimistic value function given that uncertainty that is then used to guide decision making. To take a similar idea for rapidly learning the optimal CVaR policy, we seek to consider the uncertainty in the distribution of outcomes possible and the resulting CVaR value. To do so, we use the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality—while to our knowledge this has not been previously used in reinforcement learning settings, it is a very useful concentration

inequality for our purposes as it provides bounds on the true cumulative distribution function (CDF) given a set of sampled outcomes. We leverage these bounds in order to compute optimistic estimates of the optimal CVaR.

Our interest is in creating empirically efficient and scalable algorithms that have a theoretically sound grounding. To that end, we introduce a new algorithm for quickly learning a CVaR policy in MDPs and show that at least in the evaluation case in tabular MDPs, this algorithm indeed produces optimistic estimates of the CVaR. We also show that it does converge eventually. We accompany the theoretical evidence with an empirical evaluation. We provide encouraging empirical results on a machine replacement task (Delage and Mannor 2010), a classic MDP where risk sensitive policies are critical, as well as a well validated simulator for type 1 diabetes (Man et al. 2014) and a simulated treatment optimization task for HIV (Ernst et al. 2006). In all cases we find a substantial benefit over simpler exploration strategies. To our knowledge this is the first algorithm that performs strategic exploration to learn good CVaR MDP policies.[1]

Background and Notation

Let X be a bounded random variable with cumulative distribution function $F(x) = \mathbb{P}[X \leq x]$. The *conditional value at risk (CVaR)* at level $\alpha \in (0, 1)$ of a random variable X is then defined as (Rockafellar, Uryasev, and others 2000):

$$\mathrm{CVaR}_\alpha(X) := \sup_\nu \left\{ \nu - \frac{1}{\alpha} \mathbb{E}[(\nu - X)^+] \right\} \quad (1)$$

We define the inverse CDF as $F^{-1}(u) = \inf\{x : F(x) \geq u\}$. It is well known that when X has a continuous distribution, $\mathrm{CVaR}_\alpha(X) = \mathbb{E}_{X \sim F}[X | X \leq F^{-1}(\alpha)]$ (Acerbi and Tasche 2002). For ease of notation we sometimes write CVaR as a function of the CDF F, $\mathrm{CVaR}_\alpha(F)$.

We are interested in the CVaR of the discounted cumulative reward in a **Markov Decision Process (MDP)**. An MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are finite state and action space, $r \sim R(s, a)$ is the reward distribution, $s' \sim P(s, a)$ is the transition kernel and $\gamma \in [0, 1)$ is the discount factor. A stationary policy π maps each state $s \in \mathcal{S}$ to a probability distribution over action space $\mathcal{A}$.

Let $\mathcal{Z}$ denote the space of distributions over returns (discounted cumulative rewards) from such an MDP, and assume that these returns are in $[V_{\min}, V_{\max}]$ almost surely, where $V_{\min} \geq 0$. We define $Z_\pi(s, a) \in \mathcal{Z}$ to be the distribution of the return of policy π with CDF $F_{Z_\pi(s,a)}$ and initial state action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ as $Z_\pi(s, a) := \mathrm{Law}_\pi \left(\sum_{t=0}^\infty \gamma^t R_t | S_0 = s, A_0 = a \right)$. RL algorithms most commonly optimize policies for expected return and explicitly learn Q-values, $Q^\pi(s, a) = \mathbb{E}[Z_\pi(s, a)]$ by applying approximate versions of Bellman backups. Instead, we are interested in other properties of the return distribution and we will build on several recently proposed algorithms that aim to learn a parametric model of the entire return distribution

instead of only its expectation. Such approaches are known as *distributional RL methods*.

Distributional Reinforcement Learning Distributional RL methods apply a sample-based approximation to distributional versions of the usual Bellman operators. For example, one can define a distributional Bellman operator (Bellemare, Dabney, and Munos 2017) as $\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ as

$$\mathcal{T}^\pi Z_\pi(s, a) \stackrel{D}{:=} R(s, a) + \gamma P^\pi Z(s, a) \quad (2)$$

where $\stackrel{D}{=}$ denotes equality in distribution, and the transition operator is defined as $P^\pi Z(s, a) \stackrel{D}{:=} Z(s', a')$ with $s' \sim P(\cdot|s, a)$, $a' \sim \pi(s)$. The optimality version $\mathcal{T}$ is similarly any $\mathcal{T} Z = \mathcal{T}^\pi Z$ where π is an optimal policy w.r.t. expected return. Note that this is not necessarily unique when there are multiple optimal policies. (Rowland et al. 2018) showed that $\mathcal{T}^\pi$ is a $\sqrt{\gamma}$-contraction in the Cramér-metric, $\bar{\ell}_2$

$$\bar{\ell}_2(Z_1, Z_2) = \sup_{s,a} \ell_2(Z_1(s, a), Z_2(s, a)) \quad (3)$$

$$= \sup_{s,a} \left(\int (F_{Z_1(s,a)}(u) - F_{Z_2(s,a)}(u))^2 du \right)^{1/2}$$

One of the canonical algorithms in distributional RL is CDRL or C51 (Bellemare, Dabney, and Munos 2017) which represent the return distribution Z^π as a discrete distribution with fixed support on N atoms $\{z_i = V_{\min} + i\Delta z : 0 \leq i < N\}, \Delta z := \frac{V_{\max} - V_{\min}}{N-1}$ the discrete distribution is parameterized as $\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^N$:

$$Z_\theta(s, a) = z_i \quad \text{w.p.} \quad p_i(s, a) = \frac{e^{\theta_i(s,a)}}{\sum_j e^{\theta_j(s,a)}}.$$

Essentially, C51 uses a sample transition (s, a, r, s') to perform an approximate Bellman backup $Z \leftarrow \Pi_{\mathcal{C}} \hat{\mathcal{T}} Z$, where $\hat{\mathcal{T}}$ is a sample-based Bellman operator and $\Pi_{\mathcal{C}}$ is a projection back onto the support of discrete distribution $\{z_0, \ldots, z_{N-1}\}$.

Optimistic Distributional Operator

In contrast to the typical RL setup where an agent tries to maximize its expected return, we seek to learn a stationary policy that maximizes the CVaR_α of the return at risk level α.[2] To find such policies quickly, we follow the optimism-in-the-face-of-uncertainty (OFU) principle and introduce optimism in our CVaR estimates to guide exploration. While adding a bonus to rewards is a popular approach for optimism in the standard expected return case (Ostrovski et al. 2017), we here follow a different approach and introduce optimism into our return estimates by shifting the empirical CDFs. Formally, consider a return distribution $Z(s, a) \in \mathcal{Z}$

---

[1]For more detailed manuscript with proofs and additional experiments refer to: https://arxiv.org/abs/1911.01546

[2]Note that the CVaR-optimal policy at any state can be non-stationary (Shapiro, Dentcheva, and Ruszczyński 2009), as it depends on the sum of rewards achieved up to that state. For simplicity, as (Dabney et al. 2018b) we instead seek a stationary policy, which will generally can be suboptimal but typically still achieve high CVaR, as observed in our experiments.
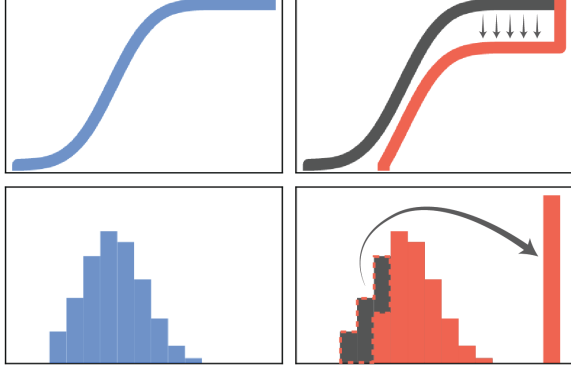
Figure 1: Top-left: Empirical CDF Top-right: The lower DKW confidence band (a shifted-down version of the empirical CDF). Bottom-left: Empirical PDF. Bottom-right: Optimistic PDF.

with CDF $F_{Z(s,a)}(x)$. We define the optimism operator $O_c : \mathcal{Z} \to \mathcal{Z}$ as

$$F_{O_c Z(s,a)}(x) = \left(F_{Z(s,a)}(x) - c\frac{\mathbf{1}\{x \in [V_{\min}, V_{\max})\}}{\sqrt{n(s,a)}} \right)^+ \tag{4}$$

where c is a constant and $(\cdot)^+$ is short for $\max\{\cdot, 0\}$. In the definition above, $n(s,a)$ is the number of times the pair (s, a) has been observed so far or an approximation such as pseudo-counts (Bellemare et al. 2016). By shifting the cumulative distribution function down, this operator essentially puts probability mass from the lower tail to the highest possible value $V_{\max}$. An illustration is provided in Figure 1. This approach to optimism is motivated by an application of the DKW-inequality to the empirical CDF. As shown recently by (Thomas and Learned-Miller 2019), this can yield tighter upper confidence bounds on the CVaR.

Theoretical Analysis

The optimistic operator introduced above operates on the entire return distribution and our algorithm introduced in the next section combines this optimistic operator to estimated return-to-go distributions. As such, it belongs to the family of distributional RL methods (Dabney et al. 2018b). These methods are a recent development and come with strong asymptotic convergence guarantees when used for *policy evaluation* in tabular MDPs (Rowland et al. 2018). Yet, finite sample guarantees such as regret or PAC bounds still remain elusive for distributional RL *policy optimization* algorithms.

A key technical challenge in proving performing bounds for distributionally robust policy optimization during RL is that convergence of the distributional Bellman optimality operator can generally not be guaranteed. Prior results have only showed that if the optimization process itself is to compute a policy which maximizes expected returns, such as Q-learning, then convergence of the distirbutional Bellman op-

timality operator is guaranteed to converge. (Rowland et al. 2018, Theorem 2). Note however that if the goal is to leverage distributional information to compute a policy to maximize something other than expected outcomes, such as a risk sensitive policy like we consider here, no prior theoretical results are known in the reinforcement learning setting to our knowledge. However, it is promising that there is some empirical evidence that one can compute risk-sensitive policies using distributional Bellman operators (Dabney et al. 2018a) which suggests that more theoretical results may be possible.

Here we take a first step towards this goal. Our primary aim in this work is to provide tools to introduce optimism into distributional return-to-go estimates to guide sample-efficient exploration for CVaR. Therefore, our theoretical analysis focuses on showing that this form of optimism does not harm convergence and is indeed a principled way to obtain optimistic CVaR estimates.

First, we prove that the optimism operator is a non-expansion in the Cramér distance. This results shows that this operator can be used with other contraction operators without negatively impacting the convergence behaviour. Specifically we can guarantee convergence with distributional Bellman backup.

Proposition 1 *For any c, the O_c operator is a non-expansion in the Cramér distance $\bar{\ell}_2$. This implies that optimistic distributional Bellman backups $O_c \mathcal{T}^\pi$ and the projected version $\Pi_{\mathcal{C}} O_c \mathcal{T}^\pi$ are $\sqrt{\gamma}$-contractions in $\bar{\ell}_2$ and iterates of these operators converge in $\bar{\ell}_2$ to a unique fixed-point.*

After establishing that our optimism operator can be used in Bellman backups without negatively impacting convergence behavior, we provide theoretical evidence that it indeed produces optimistic CVaR estimates. Consider here batch policy evaluation in MDPs M with finite state- and action-spaces. Assume that we have collected a fixed number of samples $n(s, a)$ (which can vary across states and actions) and build an empirical model $\hat{M}$ of the MDP. For any policy π, let $\hat{\mathcal{T}}^\pi$ denote the distributional Bellman operator in this empirical MDP. Then we indeed achieve optimistic estimates by the following result:

Theorem 2 *Let the shift parameter in the optimistic operator be $c \geq \sqrt{(1 + 4|\mathcal{S}|)\ln(4/\delta)}$. Then with probability at least $1 - \delta$, the iterates $\mathrm{CVaR}_\alpha((O_c \hat{\mathcal{T}}^\pi)^m Z_0)$ converges for any risk level α and initial $Z_0 \in \mathcal{Z}$ to an optimistic estimate of the policy's conditional value at risk. That is, with probability at least $1 - \delta$,*

$$\mathrm{CVaR}_\alpha((O_c \hat{\mathcal{T}}^\pi)^\infty Z_0(s, a)) \geq \mathrm{CVaR}_\alpha(Z_\pi(s, a)) \quad \forall s, a.$$

Notice that relatively little optimism c is sufficient to absorb any sampling error in the model and ensure optimistic CVaR estimates. Most importantly, there is no dependency in c on the risk level or the range of the return which indicates that this parameter is likely easy to tune empirically.

Algorithm

In the policy evaluation case where we would like to compute optimistic estimates of the CVaR of a given observed

policy π, our algorithm essentially performs an approximate version of the optimistic Bellman update $O_c \mathcal{T}^\pi$ where $\mathcal{T}^\pi$ is the distributional Bellman operator. For the control case where we would like to learn a policy that maximizes CVaR, we instead define a distributional Bellman optimality operator $\mathcal{T}_\alpha$. Analogous to prior work (Bellemare, Dabney, and Munos 2017), $\mathcal{T}_\alpha$ is any operator that satisfies $\mathcal{T}_\alpha Z = \mathcal{T}^\pi Z$ for some policy π that is greedy w.r.t. CVaR at level α. Our algorithm then performs an approximate version of the optimistic Bellman backup $O_c \mathcal{T}_\alpha$, shown in Algorithm 1.

The main structure of our algorithm resembles categorical distributional reinforcement learning (C51) (Bellemare, Dabney, and Munos 2017). In a similar way, our algorithm also maintains a return distribution estimate for each state-action pair, represented as a set of N weights $p_i(s, a)$ for $i \in [N]$. These weights represent a discrete distribution with outcomes at N equally spaced locations $z_0 < z_1 < \cdots < z_{N-1}$, each $\Delta z = \frac{V_{\max} - V_{\min}}{N-1}$ apart. The current probability assigned to outcome z_i in (s, a) is denoted by $p_i(s, a)$, where the atom probabilities $p_{1:N}(s, a)$ are given by a differentiable model such as a neural network, similar to C51. Note that other parameterized representations of the weights (Bellemare, Dabney, and Munos 2017) are straightforward to incorporate.

The main differences between Algorithm 1 and existing distributional RL algorithms (e.g. C51) are as follows: We first apply an optimism operator to our successor distribution $F_{Z(s_{t+1}, a)}$ (Lines 4–6) to form an optimistic CDF $\tilde{F}_{Z(s_{t+1}, a)}$ for all actions $a \in \mathcal{A}$. This operator should encourage exploring actions that might lead to higher CVaR policies for our input α. These optimistic CDFs are also used to decide on the successor action in the control setting (Line 8). Then, similar to C51 we apply the Bellman operator $\hat{\mathcal{T}} z_i$ for $i \in [N]$ and distribute the probability of $\tilde{p}_i$ to the immediate neighbours of $\hat{\mathcal{T}} z_i$, where we calculate the probability mass $\tilde{p}_i$ with the optimistic CDF $\tilde{F}_{Z(s_{t+1}, a^\star)}$ (Line 13).

Following (Bellemare, Dabney, and Munos 2017), we train this model using the cross-entropy loss, which for a particular state transition at time t is

$$\mathcal{L} = -\sum_{j=0}^{N-1} m_j \log p_j(s_t, a_t) \tag{5}$$

where $m_{0:N-1}$ are the weights of the target distribution computed in Lines 11–18 in Algorithm 1. In the tabular setting we can directly update the probability mass p_j by

$$p_j(s_t, a_t) = (1 - \beta) p_j(s_t, a_t) + \beta m_j(s_t, a_t)$$

where β is the learning rate.

In tabular settings, the counts $n(s, a)$ can be directly stored and used; however, this is not the case in continuous settings. For this reason, we adopt the pseudo-count estimation method proposed by (Ostrovski et al. 2017) and replace $n(s, a)$ by a pseudo-count $\hat{n}_t(s, a)$ in the optimistic distributional operator (Equation 4). Let ρ be a density model and $\rho_t(s, a)$ the probability assigned to the state action pair (s, a) by the model after t training steps. The prediction gain PG of ρ is defined

$$PG_t(s, a) = \log \rho'_t(s, a) - \log \rho_t(s, a)$$

Algorithm 1: CVaR-MDP

Input: Parameters: γ, risk level $\alpha \in (0, 1)$, $c \geq 0$, density model ρ,

1 **for** *t=1, …* **do**
2   Observe transition s_t, a_t, r_t, s_{t+1};
3   **for** $a' \in \mathcal{A}$ **do**
     /* emp. return CDF for (s_{t+1}, a') */
4      $\hat{F}^{a'}(x) := \sum_{j=0}^{N-1} p_j(s_{t+1}, a') \mathbf{1}\{x \geq z_j\}$;
     /* Pseudo-counts */
5      $\hat{n} = \frac{1}{\exp(\kappa t^{-1/2} \alpha (\nabla \log \rho_\theta(s_{t+1}, a'))^2) - 1}$
     /* Optimistic CDF */
6      $\tilde{F}^{a'}(x) := \left[\hat{F}^{a'}(x) - \frac{c \mathbf{1}\{x \in [V_{\min}, V_{\max}]\}}{\sqrt{\hat{n}}} \right]^+$;
7   **if** *Control* **then**
8      $a^\star \leftarrow \arg\max_{a \in \mathcal{A}} \mathrm{CVaR}_\alpha(\tilde{F}^a)$
9   **if** *Evaluation* **then**
10      $a^\star \sim \pi(. | s_{t+1})$
11    $m_i = 0$ for $i \in \{0, \ldots, N-1\}$;
12   **for** $j \in 0, \ldots, N-1$ **do**
     /* optimistic PDF */
13      $\tilde{p}_j \leftarrow \tilde{F}^{a^\star}\left(z_j + \frac{\Delta z}{2}\right) - \tilde{F}^{a^\star}\left(z_j - \frac{\Delta z}{2}\right)$;
     /* Project on support of $\{z_i\}$ */
14      $\hat{\mathcal{T}} z_j \leftarrow [r_t + \gamma z_j]_{V_{\min}}^{V_{\max}}$;
     /* Distribute prob. of $\hat{\mathcal{T}}(z_j)$ */
15      $b_j \leftarrow (\hat{\mathcal{T}} z_j - V_{\min}) / (\Delta z)$;
16      $l \leftarrow \lfloor b_j \rfloor; \quad u \leftarrow \lceil b_j \rceil$;
17      $m_l \leftarrow m_l + \tilde{p}_j (u - b_j)$;
18      $m_u \leftarrow m_u + \tilde{p}_j (b_j - l)$;
19   Update return weights $p_{1:N}$ by optimization step on cross-entropy loss
     $\mathcal{L} = -\sum_{j=0}^{N-1} m_j \log p_j(s_t, a_t)$;
   /* Take next action */
20    $a_{t+1} \leftarrow a^\star$;
21   Update density model ρ with additional observation of (s_{t+1}, a_{t+1});

Where $\rho'_t(s, a)$ is the probability assigned to (s, a) if it were trained on that same (s, a) one more time. Now we define the pseudo count of (s, a) as

$$\hat{n}_t(s, a) = (\exp(\kappa t^{-\frac{1}{2}} (PG_t(s, a))_+) - 1)^{-1} \tag{6}$$

where κ is a constant hyper-parameter, and $(PG(s, a))_+$ thresholds the value of the prediction gain at 0.

Our setting differs from (Ostrovski et al. 2017) in the sense that we have to compute the count before taking the action a. A naive way would be to try all actions and train the model to compute the counts but this method is slow and requires the environment to support an undo action. Instead, we can estimate PG for all actions as follows. Consider the density model parametrized by θ, $\rho(s, a; \theta)$. After observing (s, a), the training step to maximize the log likelihood will update the parameters by $\theta' = \theta + \alpha \nabla_\theta \log \rho(s, a; \theta)$, where α is the learning rate. So we can approximate the new log

probability using a first-order Taylor expansion

$$\log \rho'_t(s, a) = \log \rho(s, a; \theta')$$
$$\approx \log \rho(s, a; \theta) + \nabla_\theta \log \rho(s, a; \theta)(\theta' - \theta)$$
$$= \log \rho(s, a; \theta) + \alpha(\nabla_\theta \log \rho(s, a; \theta))^2$$

This calculation suggests that the prediction gain can be estimated just by computing the gradient of the log likelihood given a state-action pair, i.e.,

$$PG(s, a) \approx \alpha(\nabla_\theta \log \rho(s, a; \theta))^2$$

As discussed in (Graves et al. 2017) this estimate of prediction gain is biased, but empirically we have found this method to perform well.

Experimental Evaluation

We validate our algorithm empirically in three simulated environments against baseline approaches.

Simulation Domains

Finance, health and operations are common areas where risk-sensitive strategies are important, and we focus on two health domains and one operations domain. Details, where omitted, are provided in the supplemental material.

Machine Replacement   Machine repair and replacement is a classic example in the risk sensitive literature, though to our knowledge no prior work has considered how to quickly learn a good risk-sensitive policy for such domains. Here we consider a minor variant of a prior setting (Delage and Mannor 2010). Specifically, as shown in Figure 2, the environment consists of a chain of n (25 in our experiments) states. There are two actions: *replace* and *don't replace*. Choosing *replace* at any state terminates the episode, while choosing *don't replace* moves the agent to the next state in the chain. At the end of the chain, choosing *don't replace* terminates the episode with a high variance cost, and choosing *replace* terminates the episode with a higher cost but lower variance. This environment is especially a challenging exploration task due to the chain structure of the MDP, as well as the high variance of the reward distributions when taking actions in the last state. Additionally in this MDP it is feasible to exactly compute the $\mathrm{CVaR}_{0.25}$-optimal policy, which allows us to compare the learned policy to the true optimal CVaR policy. Note here that the optimal policy for maximizing $\mathrm{CVaR}_{0.25}$ is to *replace* on the final state in the chain to avoid the high variance alternative; in contrast, the optimal policy for expected return always chooses *don't replace*.

HIV Treatment   In order to test our algorithm on a larger continuous state space, we leverage an HIV Treatment simulator. The environment is based on the implementation by (Geramifard et al. 2015) of the physical model described in (Ernst et al. 2006). The patient state is represented as a 6-dimensional continuous vector and the reward is a function of number of free HIV viruses, immune response of the body to HIV, and side effects. There are four actions, each determining which drugs are administered for the next 20 day
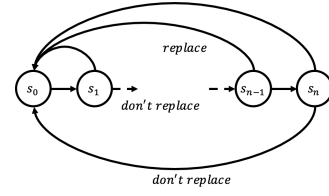

Figure 2: Machine Replacement: This environment consists of a chain of n states, each affording two actions: *replace* and *don't replace*.
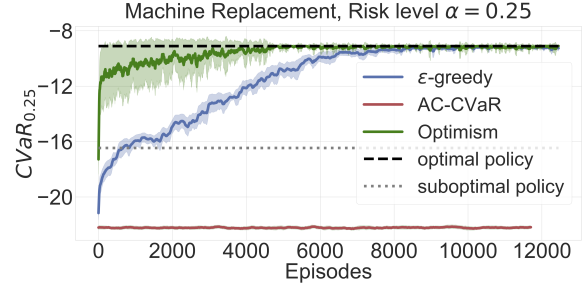

Figure 3: Machine Replacement: The thick grey dashed line is the $\mathrm{CVaR}_{0.25}$-optimal policy. The thin dashed lines labeled as the suboptimal policy is the optimal expectation-maximizing policy. The shaded area shows the 95% confidence intervals.

period: Reverse Transcriptase Inhibitors (RTI), Protease Inhibitors (PI), neither, or both. There are 50 time steps in total per episode, for a total of 1000 days. We chose here a larger number of days per time step compared to the typical setup (200 steps of 5 days each) to facilitate faster experimentation. This design choice also makes the exploration task harder, since taking one wrong action can drastically destabilize a patient's trajectory. The original proposed model was deterministic, which makes the CVaR policy identical to the policy optimizing the expected value. Such simulators are rarely a perfect proxy for real systems, and in our setting we add Gaussian noise $\sim \mathcal{N}(0, 0.01)$ to the efficacy of each drug (RTI: ϵ_1 and PI: ϵ_2 in (Ernst et al. 2006)). This change necessitates risk-sensitive policies in this environment.

Diabetes 1 Treatment   Patients with type 1 diabetes regulate their blood glucose level with insulin in order to avoid hypoglycemia or hyperglycemia (very low or very high blood glucose level, respectively). Recently, a simulator has been introduced (Man et al. 2014) that is an open source version of a simulator that was approved by the FDA as a substitute for certain pre-clinical trials. The state is continuous-valued vector of the current blood glucose level and the amount of carbohydrate intake (through food). The action space is discretized into 6 levels of a bolus insulin injection. The reward function is defined similar to the prior work
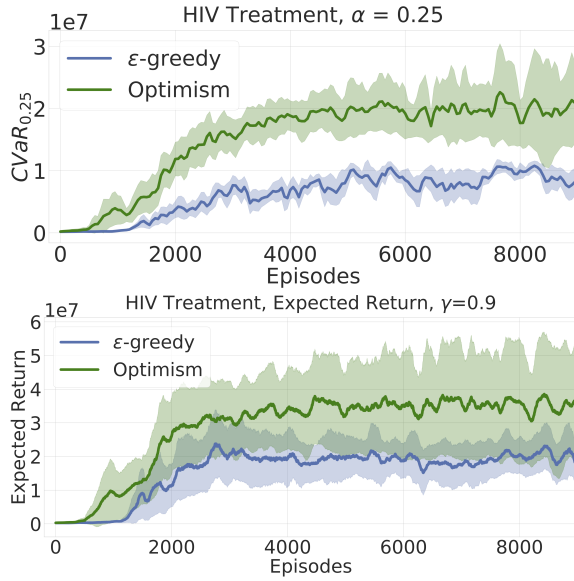
Figure 4: Comparison of our approach against an ϵ-greedy baseline. Both models were trained to optimize the $\mathrm{CVaR}_{0.25}$ of the return on a stochastic version of the HIV simulator (Ernst et al. 2006). Top: Objective $\mathrm{CVaR}_{0.25}$; Bottom: Discounted expected return of the same policies as in top plot.

(Bastani 2014) as following:

$$r(bg) = \begin{cases} -\frac{(bg'-6)^2}{5} & if\ bg' < 6 \\ -\frac{(bg'-6)^2}{10} & if\ bg' \geq 6 \end{cases}$$

Where $bg' = bg/18.018018$ which is the estimate of bg (blood glucose) in mmol/L.

Additionally we inject two source of stochasticity into the taken action: First, we add Gaussian noise $\mathcal{N}(0,1)$ to the action. Second, we delay the time of the injection by at most 5 steps, where the probability of injection at time t is higher than time $t+i, i \geq 1$ following the power law. Each simulation lasts for 200 steps, during which a patient eats five meals. The agent chooses an action after each meal, and after the 200 steps each patient resets to its initial state.

This domain also readily offers a suite of related tasks, since the environment simulates 30 patients with slightly different dynamics. Tuning hyper-parameters on the same task can be misleading (Henderson et al. 2018), as is the case in our two previous benchmarks. In this setting we tune baselines and our method on one patient, and test the performance on different patients.

Baselines and Experimental Setup

The majority of prior risk-sensitive RL work has not focused on efficient exploration, and there has been very little deep distributional RL work focused on risk sensitivity. Our key contribution is to evaluate the impact of more strategic exploration on the efficiency with which a risk-sensitive policy can be learned. Therefore we compare to two illustrative approaches:

1. ϵ-greedy CVaR: In this benchmark we use the same algorithm, except we do not introduce an optimism operator, instead using an ϵ-greedy approach for exploration. This benchmark can be viewed as analogous to the distributional RL methods of C51 (Bellemare, Dabney, and Munos 2017) (if the computed policy had optimized for CVaR instead of expected reward) or the risk-sensitive implicit quantile regression (Dabney et al. 2018a) approach (but using the alternate projection step, as done in C51).

2. CVaR-AC: An actor-critic method proposed by (Chow and Ghavamzadeh 2014) that maximizes the expected return while satisfying an inequality constraint on the CVaR. This method relies on the stochasticity of the policy for exploration.

Note that a comparison to an expectation maximizing algorithm is uninformative since such approaches are maximizing different (non-risk-sensitive) objectives.

All of these algorithms use hyperparameters, and it is well recognized that ϵ-greedy algorithms can often perform quite well if their hyperparameters are well-tuned. To provide a fair comparison, we evaluated across a number of schedules for reducing the ϵ parameter, and a small set of parameters (4-7) for the optimism value c for our method. We used the specification described in Appendix C of (Chow and Ghavamzadeh 2014) for CVaR-AC.

The system architectures used in continuous settings are identical for Baseline 1 (ϵ-greedy) and our method. This consists of 2 hidden layers of size 32 with ReLU activation for Diabetes 1 Treatment, and 4 hidden layers of size 128 with ReLU activation for HIV Treatment, both followed by a softmax layer for each action. The density model is a realNVP (Dinh, Sohl-Dickstein, and Bengio 2016) with 3 hidden layers each of size 64.

All results are averaged over 10 runs and we report 95% confidence intervals. We report the performance of ϵ-greedy at evaluation time (setting $\epsilon = 0$), which is the best performance of ϵ-greedy.

For the Diabetes Treatment domain, hyperparameters are optimized only on `adult#001`. We then report results of the methods using those hyperparameters on `adult#003`, `adult#004` and `adult#005`.

Results and Discussion

Results on machine replacement environment (Figure 3), HIV Treatment (Figure 4) and Diabetes 1 Treatment (Figure 5) all show our optimistic algorithm achieves better performance much faster than the baselines.

In Machine Replacement (Figure 3) we see that our method quickly converges to the optimal CVaR performance. Unfortunately despite our best efforts, our implementation of CVaR-AC did not perform well even on the simplest environment, so we did not show the performance of this method on other environments. One challenge here is that CVaR-AC has a significant number of hyperparameters, including 3 different learning rates schedule for the optimization process, initial Lagrange multipliers and the kernel functions.
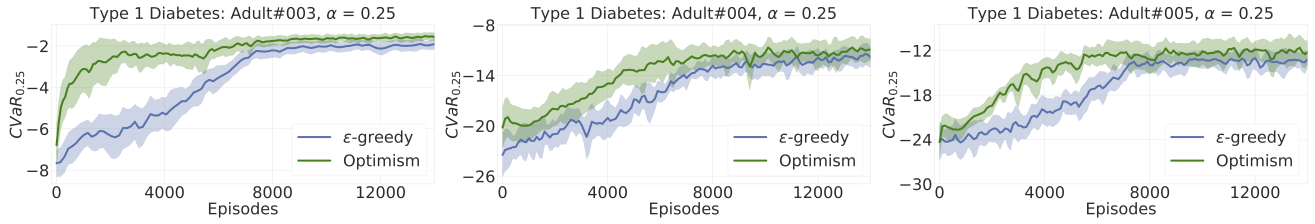
Figure 5: Type 1 diabetes simulator: $CVaR_{0.25}$ for three different adults. Plots are averaged over 10 runs with 95% CI.

	ϵ-greedy	CVaR-MDP
Adult#003	$11.2\% \pm 3.6\%$	$\mathbf{4.2\%} \pm \mathbf{2.3\%}$
Adult#004	$2.3\% \pm 0.3\%$	$\mathbf{1.4\%} \pm \mathbf{0.6\%}$
Adult#005	$3.3\% \pm 0.3\%$	$\mathbf{1.7\%} \pm \mathbf{0.6\%}$

Figure 6: Type 1 Diabetes simulator, percent of episodes where patients experienced a severe medical condition (hypoglycemia or hyperglycemia), averaged across 10 runs

In the HIV Treatment we also see a clear and substantial benefit to our optimistic approach over the baseline ϵ-greedy approach (Figure 4).

Figure 5 is particularly encouraging, as it shows the results for the diabetes simulator across 3 patients, where the hyperparameters were fixed after optimizing for a separate patient. Since in real settings it would be commonly necessary to fix the hyperparameters in advance, this result provides a nice demonstration that the optimistic approach can consistently equal or significantly improve over an ϵ-greedy policy in related settings, similar to the well known results in Atari in which hyperparameters are optimized for one game and then used for multiple others.

"Safer" Exploration. Our primary contribution is a new algorithm to learn risk-sensitive policies quickly, with less data. However, an interesting side benefit of such a method might be that the number of extremely poor outcomes experienced over time may also be reduced, not due to explicitly prioritizing a form of safe exploration, but because our algorithm may enable a faster convergence to a safe policy. To evaluate this, we consider a risk measure proposed by (Clarke and Kovatchev 2009), which quantifies the risk of a severe medical condition based on how close their glucose level is to hypoglycemia (blood glucose, ≤ 3.9 mmol/l) and hyperglycemia (blood glucose, ≥ 10 mmol/l). Table 6 shows the fraction of episodes in which each patient experienced a severely poor outcome for each algorithm while learning. Optimism-based exploration approximately halves the number of episodes with severely poor outcomes, highlighting a side benefit of our optimistic approach of more quickly learning a good safe policy.

Related Work

Optimizing policies for risk sensitivity in MDPs has been long studied, with policy gradient (Tamar et al. 2015), ac-

tor critic (Tamar and Mannor 2013) and TD methods (Sato, Kimura, and Kobayashi 2001). While most of this work considers mean-variance trade objectives, (Chow et al. 2015) establish a connection between a optimizing CVaR and robustness to modeling errors, presenting a value iteration algorithm. In contrast, we do not assume access to transition and rewards models. (Chow and Ghavamzadeh 2014) present a policy gradient and actor-critic algorithm for an expectation-maximizing objective with a CVaR constraint. None of these works considers systematic exploration but rely on heuristics such as ϵ-greedy or on the stochasticity of the policy for exploration. Instead, we focus on how to explore systematically to find a good CVaR-policy.

Our work builds upon recent advances on distributional RL (Bellemare, Dabney, and Munos 2017; Rowland et al. 2018; Dabney et al. 2018b) which are still concerned with optimizing expected return. Notably, (Dabney et al. 2018a) aims to train risk-averse and risk-seeking agents, but does not address the exploration problem or attempts to find optimal policies quickly.

(Dilokthanakul and Shanahan 2018) uses risk-averse objectives to guide exploration for good performance w.r.t. expected return. (Moerland, Broekens, and Jonker 2018) leverages the return distribution learned in distributional RL as a means for optimism in deterministic environments. (Mavrin et al. 2019) follow a similar pattern but can handle stochastic environments by disentangling intrinsic and parametric uncertainty. While they also evaluate the policy that picks the VaR-greedy action in one experiment, their algorithm still optimizes expected return during learning. In general, these approaches are fundamentally different from ours which learns CVaR policies in stochastic environments efficiently by introducing optimism *into* the learned return distribution.

Acknowledgments

References

Acerbi, C., and Tasche, D. 2002. On the coherence of expected shortfall. *Journal of Banking & Finance* 26(7):1487–1503.

Artzner, P.; Delbaen, F.; Eber, J.-M.; and Heath, D. 1999. Coherent measures of risk. *Mathematical finance* 9(3):203–228.

Bastani, M. 2014. Model-free intelligent diabetes management using machine learning.

Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 1471–1479.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 449–458. JMLR. org.

Berkenkamp, F.; Turchetta, M.; Schoellig, A.; and Krause, A. 2017. Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, 908–918.

Brafman, R. I., and Tennenholtz, M. 2002. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3(Oct):213–231.

Chow, Y., and Ghavamzadeh, M. 2014. Algorithms for cvar optimization in mdps. In *Advances in neural information processing systems*, 3509–3517.

Chow, Y.; Tamar, A.; Mannor, S.; and Pavone, M. 2015. Risk-sensitive and robust decision-making: a cvar optimization approach. In *Advances in Neural Information Processing Systems*, 1522–1530.

Clarke, W., and Kovatchev, B. 2009. Statistical tools to analyze continuous glucose monitor data. *Diabetes technology & therapeutics* 11(S1):S–45.

Dabney, W.; Ostrovski, G.; Silver, D.; and Munos, R. 2018a. Implicit quantile networks for distributional reinforcement learning. *arXiv preprint arXiv:1806.06923*.

Dabney, W.; Rowland, M.; Bellemare, M. G.; and Munos, R. 2018b. Distributional reinforcement learning with quantile regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Dann, C.; Li, L.; Wei, W.; and Brunskill, E. 2018. Policy certificates: Towards accountable reinforcement learning. *arXiv preprint arXiv:1811.03056*.

Delage, E., and Mannor, S. 2010. Percentile optimization for markov decision processes with parameter uncertainty. *Operations research* 58(1):203–213.

Dilokthanakul, N., and Shanahan, M. 2018. Deep reinforcement learning with risk-seeking exploration. In *International Conference on Simulation of Adaptive Behavior*, 201–211. Springer.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Ernst, D.; Stan, G.-B.; Goncalves, J.; and Wehenkel, L. 2006. Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 667–672. IEEE.

Geramifard, A.; Dann, C.; Klein, R. H.; Dabney, W.; and How, J. P. 2015. Rlpy: a value-function-based reinforcement learning framework for education and research. *Journal of Machine Learning Research* 16(46):1573–1578.

Graves, A.; Bellemare, M. G.; Menick, J.; Munos, R.; and Kavukcuoglu, K. 2017. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1311–1320. JMLR. org.

Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; and Meger, D. 2018. Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jaksch, T.; Ortner, R.; and Auer, P. 2010. Near-optimal regret bounds for reinforcement learning.

Koller, T.; Berkenkamp, F.; Turchetta, M.; and Krause, A. 2018. Learning-based model predictive control for safe exploration. In

2018 IEEE Conference on Decision and Control (CDC), 6059–6066. IEEE.

Man, C. D.; Micheletto, F.; Lv, D.; Breton, M.; Kovatchev, B.; and Cobelli, C. 2014. The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology* 8(1):26–34.

Mavrin, B.; Yao, H.; Kong, L.; Wu, K.; and Yu, Y. 2019. Distributional reinforcement learning for efficient exploration. In *International Conference on Machine Learning*, 4424–4434.

Moerland, T. M.; Broekens, J.; and Jonker, C. M. 2018. The potential of the return distribution for exploration in rl. *arXiv preprint arXiv:1806.04242*.

Ostrovski, G.; Bellemare, M. G.; van den Oord, A.; and Munos, R. 2017. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2721–2730. JMLR. org.

Rockafellar, R. T.; Uryasev, S.; et al. 2000. Optimization of conditional value-at-risk. *Journal of risk* 2:21–42.

Rowland, M.; Bellemare, M. G.; Dabney, W.; Munos, R.; and Teh, Y. W. 2018. An analysis of categorical distributional reinforcement learning. *arXiv preprint arXiv:1802.08163*.

Sato, M.; Kimura, H.; and Kobayashi, S. 2001. Td algorithm for the variance of return and mean-variance reinforcement learning. *Transactions of the Japanese Society for Artificial Intelligence* 16(3):353–362.

Shapiro, A.; Dentcheva, D.; and Ruszczyński, A. 2009. *Lectures on stochastic programming: modeling and theory*. SIAM.

Strehl, A. L., and Littman, M. L. 2008. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences* 74(8):1309–1331.

Tamar, A., and Mannor, S. 2013. Variance adjusted actor critic algorithms. *arXiv preprint arXiv:1310.3697*.

Tamar, A.; Chow, Y.; Ghavamzadeh, M.; and Mannor, S. 2015. Policy gradient for coherent risk measures. In *Advances in Neural Information Processing Systems*, 1468–1476.

Tamar, A.; Glassner, Y.; and Mannor, S. 2015. Optimizing the cvar via sampling. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Thomas, P., and Learned-Miller, E. 2019. Concentration inequalities for conditional value at risk. In *International Conference on Machine Learning*, 6225–6233.

Zhu, S., and Fukushima, M. 2009. Worst-case conditional value-at-risk with application to robust portfolio management. *Operations research* 57(5):1155–1168.