Random Barrier Avoidance with Risk-Constrained MDPs

Lucas Berry

April 21st 2017

1 Introduction

When making decisions one is often confronted with the notion of risk. These situations appear in everyday life, choosing whether or not to go to a party that your ex might attend, to sneak out of the house when your mom might catch you or to speed when you are late for work are all examples of situations that involve a potential risky outcome. Catastrophic events have high cost and usually occur seldomly, making them hard to capture in one's model. Researchers have often tried to develop models or policies that incorporate some penalization for riskiness and these results have multiple applications, i.e. financial mathematics, medical decision making and robotics. In each one of these domains failure comes at a high cost: bankruptcy or economic collapse, death and broken robots. Many problems can be framed into a sequential decision making problem such as a Markov decision process (MDP).

The typical episodic reinforcement learning problem involves optimizing a risk-neutral return as in Sutton and Barto (2017),

$$\max E\left[G_T | s_0, \pi\right] \tag{1}$$

where $G_T = \sum_{i=0}^{T} \gamma^i R_i$ is the discounted return, T is the terminal time, s_0 is the initial state and π is a policy that determines which actions are taken in according to the current state. The problem is stated in such a way that disregards rare high cost events, when maximizing the expected returns there is no restriction set on the tail of the resulting distribution. When taking a risk-neutral approach in finance one can end up on the street with no money. This led to the evolution of different measures of risk, from Markowitz (1968) analyzing a mean variance portfolio optimization, to the more common risk measures of today in Acerbi and Tasche (2002). The analysis of high cost rare events has a long history in finance and many of these ideas can be applied to other areas like medicine, Shen et al. (2014). Risk sensitive MDPs have been studied with different risk measures, one of the earliest papers being Howard and Matheson (1972) where they used an exponential risk metric $\frac{1}{\gamma}E[exp(\gamma G_T)]$. In their formulation, γ controlled the risk aversion. Today many risk sensitive algorithms have followed the methods from finance: Prashanth and Ghavamzadeh (2013) and Chow et al. (2015).

This report analyzes the algorithms developed in Chow et al. (2015) and applies them to a robotic problem of avoiding a random barrier. Most of the literature focusing on the application of risk sensitive MDPs have focused on financial applications, but robotics poses some similar high cost events that an engineer would want to avoid. Some examples exist of risk sensitive methods being applied to robotic problems, Carpin et al. (2016) and Kahn et al. (2017), but many roboticists seem to gravitate towards maximizing the risk-neutral expected return. The addition of the risk constraint should prevent the agent from taking actions that have high enough probabilities of incurring a high cost.

The report is organized as follows. In Section 2 the problem will be set up with the appropriate notation. Then the algorithms in Chow et al. (2015) and their derivation will be presented in Section 3. An explanation of the environment will be provided in Section 4, with the results from the risk-constrained algorithms compared to the appropriate algorithms from Sutton and Barto (2017). Then the report will conclude with some remarks and possible ideas for further work in Section 5.

2 Problem Formulation

The problem follows a standard Markov decision process and this section begins by defining important functions, variables and values. The notation will follow Sutton and Barto (2017).

2.1 Notation

A finite MDP can be defined as s tuple $M = (S, A, R, P, P_0)$ where the state and action spaces are defined as $S = \{1, ..., n\}$ and $A = \{1, ..., m\}$ respectively, $R(s, a)$ refers to the return of taking action a in state s, $P(s'|s, a)$ is the transition probability and $P_0(s)$ are the initial state probabilities. Let $\gamma \in [0, 1]$ denote the discount factor, where lower values signify that the agent cares more for immediate rewards whereas larger values imply that the agent focuses more on rewards received at later states. Chow et al. (2015) derive algorithms for policy gradient methods using a softmax policy. Let θ be vector of parameters and π be the policy, then

$$\pi(a|s, \theta) = \frac{\exp(\theta^T \phi(s, a))}{\sum_{b=1}^{m} \exp(\theta^T \phi(s, b))},$$

where $\phi(s, a)$ are features attached to state s and action a. Using a softmax formulation of the policy has the advantage of approaching determinism as opposed to ϵ-greedy, which always has the probability of choosing the non optimal strategy. Policy gradient also has the advantage of injecting prior knowledge into the initial policy and being able to handle continuous action spaces. Note that all demonstrations and calculations were done using a finite action space. Two important functions when deriving algorithms for MDPs are the state value function and the state action value function, which can be stated as

$$v^{\pi_\theta}(s) = E[G_T|s, \theta] \qquad q^{\pi_\theta}(s, a) = E[G_T|s, a, \theta].$$

Two common risk measures were considered as constraints in Chow et al. (2015). Let X denote a random variable, thus value at risk (VaR) can be defined as,

$$VaR_\alpha(X) := \inf\{x|\mathbb{P}(X \leq x) \geq \alpha\},$$

where $\alpha \in [0,1]$ and $x \in \mathbb{R}$. Thus VaR is a quantile and α controls the section of the tail interested in. VaR is a very popular risk measure even though it is not coherent, Artzner et al. (1999). Let r denote a risk measure. A risk measure is considered coherent if it satisfies the following 4 conditions,

1. Subadditivity: for any pair of random variables, X_1 and X_2
$$r(X_1 + X_2) \leq r(X_1) + r(X_2).$$
2. Monotonicity: $\forall\ X_1 > X_2$
$$r(X_1) \geq r(X_2)$$
3. Homogeneity: for any constant a
$$r(aX_1) = ar(X_1)$$
4. Translational invariance: for any constant a
$$r(X_1 + a) = r(X_1) + a.$$

VaR is not a coherent risk measure as it fails the subadditive condition. Therefore people have been using conditional value at risk ($CVaR$) more, as it is a coherent risk measure. Note that $CVaR$ goes by many names, average value at risk ($AVaR$) and expected shortfall (ES) to name a few. $CVaR$ is defined as,

$$CVaR_\alpha(X) := E\left[X | X \geq VaR_\alpha(X)\right].$$

Thus CVaR is an expectation on the tail of the distribution which implies $CVaR_\alpha(X) \geq VaR_\alpha(X)$. Rockafellar and Uryasev (2000) showed that $CVaR$ can be rewritten as,

$$CVaR_\alpha(X) = \min_{\nu \in \mathbb{R}} \left\{ \nu + \frac{1}{1-\alpha} E\left[(X-\nu)^+\right] \right\},$$

where $x^+ = \max(0, x)$. When optimizing a function that involves $CVaR$ it is more convenient to use the second form.

2.2 Risk-Constrained MDP

Using the described notation, the risk sensitive MDP problem can be stated as,

$$\min_\theta v_{\pi_\theta}(s_0) \quad \text{subject to} \quad CVaR_\alpha(G_T | s_0, \theta) \leq \beta, \tag{2}$$

which can be restated as,

$$\min_\theta v_{\pi_\theta}(s_0) \quad \text{subject to} \quad H_\alpha(G_T, \nu | s_0, \theta) \leq \beta, \tag{3}$$

where

$$H_\alpha(Z, \nu) := \nu + \frac{1}{1-\alpha} E\left[(Z-\nu)^+\right].$$

Note that unlike Sutton and Barto (2017) the optimization problem has been rephrased as a min which can easily be adapted to max. This formulation follows Chow et al. (2015). Unlike a risk neutral approach, formulating the problem in this fashion should protect against unacceptable high cost events, though the practitioner is forced to choose what is unacceptable, i.e. β. The problem can be formulated with a VaR constraint as well, which is also known as chance-constrained optimization. The problem then becomes,

$$\min_{\theta} v_{\pi_\theta}(s_0) \quad \text{subject to} \quad \mathbb{P}(G_T \geq \beta | s_0, \theta) \leq 1 - \alpha. \tag{4}$$

Note that here VaR has been rewritten.

To solve (3) one can use a Lagrangian relaxation procedure similar to the one described in Bertsekas (1999). Thus the optimization can be stated as max min:

$$\max_{\lambda \geq 0} \min_{\theta, \nu} \left(L(\nu, \lambda, \theta) := v_{\pi_\theta}(s_0) + \lambda \big[H_\alpha(G_T, \nu | s_0, \theta) - \beta \big] \right), \tag{5}$$

where λ is the Lagrange multiplier. Each algorithm uses (5) to derive gradients and proceeds in the appropriate direction. Note that (5) was adapted for the VaR constrained case.

3 Algorithms

This section will present the policy gradient and actor-critic algorithms described in Chow et al. (2015). The first such algorithm has episodic updates and thus is Monte Carlo in style. It relies on three gradients,

$$\nabla_\theta L(\nu, \lambda, \theta) = \nabla_\theta v_{\pi_\theta}(s_0) + \frac{\lambda}{1-\alpha} \nabla_\theta E\left[(G_T - \nu)^+ \right] \tag{6}$$

$$\partial_\nu L(\nu, \lambda, \theta) = \lambda \left[1 + \frac{1}{1-\alpha} \partial_\nu E\left[(G_T - \nu)^+ \right] \right] \tag{7}$$

$$\nabla_\lambda L(\nu, \lambda, \theta) = \nu + \frac{1}{1-\alpha} E\left[(G_T - \nu)^+ \right] - \beta \tag{8}$$

With the three gradients it is necessary to define three step sizes that decay at appropriate rates. Let $\zeta_1(i)$, $\zeta_2(i)$ and $\zeta_3(i)$ denote the step sizes. Their schedules must satisfy

$$\sum_k \zeta_1(i) = \sum_k \zeta_2(i) = \sum_k \zeta_3(i) = \infty,$$
$$\sum_k \zeta_1(i)^2, \sum_k \zeta_2(i)^2, \sum_k \zeta_3(i)^2 < \infty$$
$$\zeta_1(i) = o(\zeta_2(i)), \quad \zeta_2(i) = o(\zeta_3(i)).$$

From this we can define the first algorithm. The steps are as follows:

1. Input a policy π, a confidence level α and a cost tolerance β

2. Then initialize the three parameters which need optimizing, θ, ν and λ.

3. for k = 0,1...: Generate N episodes/trajectories by starting at s_0 and following the current policy θ_k, $\xi_{j,k} = \{s_0, a_0, r_1, ...s_T\}$. After Generating N episodes apply the following updates:

$$\theta_{k+1} = \Gamma_\Theta \Bigg[\theta_k - \zeta_2(k)\Bigg(\frac{1}{N}\sum_{j=1}^{N} \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})\big|_{\theta=\theta_k} G_T(j,k) +$$

$$\frac{\lambda_k}{(1-\alpha)N}\sum_{j=1}^{N} \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k})\big|_{\theta=\theta_k} (G_T(j,k) - \nu_k\})\mathbf{1}\{G_T(j,k) \geq \nu_k\}\Bigg)\Bigg],$$

$$\nu_{k+1} = \Gamma_\mathcal{N}\left[\nu_k - \zeta_3(k)\left(\lambda_k - \frac{\lambda_k}{(1-\alpha)N}\sum_{j=1}^{N}\mathbf{1}\{G_T(j,k) \geq \nu_k\}\right)\right],$$

$$\lambda_{k+1} = \Gamma_\Lambda\Bigg[\lambda_k - \zeta_1(k)\Bigg(\nu_k - \beta + \frac{1}{(1-\alpha)N}\sum_{j=1}^{N}(G_T(j,k) - \nu_k)\mathbf{1}\{G_T(j,k) \geq \nu_k\}\Bigg)\Bigg],$$

where $\Gamma_\Theta(\theta) = \text{argmin}_{\hat{\theta}\in\Theta} ||\theta - \hat{\theta}||_2^2$. Please note the values for the weights are bounded by Θ. This operator is similar for $\Gamma_\Lambda(\lambda)$ and $\Gamma_\mathcal{N}(\nu)$ whose values are also bounded. Also check if $|\lambda_k - \lambda_{max}| \leq \epsilon$ if so set $\lambda_{max} \leftarrow 2\lambda$.

This algorithm guarantees convergence to a local saddle point of $L(\nu, \lambda, \theta)$, see Chow et al. (2015).

The problem space must be changed before moving to the actor critic $CVaR$ algorithm. Let $\bar{M} = (\bar{S}, \bar{A}, \bar{R}, \bar{P}, \bar{P}_0)$ denote the augmented MDP where $\bar{S} = S\text{x}X$, $\bar{A} = A$, $\bar{P}_0(s,x) = P_0(s)\mathbf{1}\{x_0 = x\}$ and

$$\bar{R}(s,x,a) = \begin{cases} \frac{\lambda(-x)^+}{1-\alpha} & \text{if} \quad s = s_T \\ R(s,a) & \text{otherwise} \end{cases}$$

$$\bar{\mathbb{P}}(s', x'|s,x,a) = \begin{cases} \mathbf{1}\{s' = s_T, x' = 0\} & \text{if} \quad s = s_T \\ \mathbb{P}(s'|s,a)\mathbf{1}\{s' = (s - C(x,a))/\gamma\} & \text{otherwise} \end{cases}$$

where X and x_0 are the finite state space and the initial state of the x part of the state in the augmented MDP. As actor critic algorithms have another weight vector, w, to denote the critic another step size parameter must be included. Here are the new step size schedule conditions,

$$\sum_k \zeta_1(i) = \sum_k \zeta_2(i) = \sum_k \zeta_3(i) = \sum_k \zeta_4(i) = \infty,$$

$$\sum_k \zeta_1(i)^2, \sum_k \zeta_2(i)^2, \sum_k \zeta_3(i)^2, \sum_k \zeta_4(i)^2 < \infty$$

$$\zeta_1(i) = o(\zeta_2(i)), \quad \zeta_2(i) = o(\zeta_3(i)), \quad \zeta_3(i) = o(\zeta_4(i)).$$

Note that also $v_{\pi_\theta} = w^T\phi(s,x)$ in the augmented matrix denotes the critic. The second algorithm is a simultaneous perturbation stochastic approximation based algorithm. The algorithm is as follows,

1. Input a policy π, a confidence level α and a cost tolerance β
2. Then initialize the three parameters which need optimizing, θ, ν and λ.
3. for k = 0,1...: draw an action according to π_θ, observe a return $\bar{R}$ and observe the next state (s_k, x_k) $\bar{\mathbb{P}}$. Apply the following updates:

$$\begin{aligned}
&\delta_k(w_k) = \bar{R}(s_k, x_k, a_k) + \gamma w_k^T \phi(s_{k+1}, x_{k+1}) - w_k^T \phi(s_k, x_k) \\
&w_{k+1} = w_k + \zeta_4(k)\delta_k(w_k)\phi(s_k, x_k) \\
&\nu_{k+1} = \Gamma_{\mathcal{N}}\left(\nu_k - \zeta_3(k) \left(\lambda_k + \frac{w_k^T \phi(s_0, \nu_k + \Delta_k) - \phi(s_0, \nu_k + \Delta_k)}{2\Delta_k} \right) \right) \\
&\theta_{k+1} = \Gamma_\Theta \left(\theta_k - \frac{\zeta_2(k)}{1-\gamma} \nabla_\theta \log \pi(a_k | s_k, x_k; \theta) \cdot \delta_k(w_k) \right) \\
&\lambda_{k+1} = \Gamma_\lambda \left(\lambda_k + \zeta_1(k)(\nu_k - \beta + \frac{1}{(1-\alpha)(1-\gamma)} \mathbf{1}\{s_k = s_T\}(-s_k)^+) \right)
\end{aligned}$$

where $\Gamma_\Theta(\theta) = \text{argmin}_{\hat{\theta}\in\Theta} ||\theta - \hat{\theta}||_2^2$ and Δ_k satisfies $\Delta_k \to 0$ as $k \to 0$ and $\sum_k (\zeta_2(k)/\Delta_k)^2 < \infty$. Please note the values for the weights are bounded by Θ. This operator is similar for $\Gamma_\Lambda(\lambda)$ and $\Gamma_{\mathcal{N}}(\nu)$ whose values are also bounded. Also check if $|\lambda_k - \lambda_{max}| \leq \epsilon$ if so set $\lambda_{max} \leftarrow 2\lambda$.

Note that for both $CVaR$ algorithms $\gamma \in (0, 1)$, though Chow et al. (2015) state there results can be extended to the case where $\gamma = 1$.

Unlike the previous two algorithms the VaR constrained algorithms apply when $\gamma = 1$. The policy gradient follows the same procedure as the first algorithm just with the following updates,

$$\theta_{k+1} = \Gamma_\Theta \left[\theta_k - \zeta_2(k) \frac{1}{N} \left(\sum_{j=1}^{N} \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k}) G_T(j,k) + \lambda_k \sum_{j=1}^{N} \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k}) \mathbf{1}\{G_T(j,k) \geq \alpha\} \right) \right],$$

$$\lambda_{k+1} = \Gamma_\Lambda \left[\lambda_k - \zeta_1(k) \left(-\beta + \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}\{G_T(j,k) \geq \alpha\} \right) \right].$$

Note that there are only two step sizes as there is one parameter, the conditions for the step sizes still need to hold. Besides that, the algorithms executes with the same procedure. The actor-critic VaR constrained algorithm had the following updates,

$$\begin{aligned}
&w_{k+1} = w_k + \zeta_3(k) \sum_{h=1}^{T} \phi(s_h, x_h) \delta_h(w_k) \\
&\theta_{k+1} = \Gamma_\Theta \left(\theta_k - \zeta_2(k) \sum_{h=0}^{T} \nabla_\theta \log \pi(a_h | s_h, x_h; \theta) \big|_{\theta=\theta_k} \cdot \delta_h(w_k) \right) \\
&\lambda_{k+1} = \Gamma_\Lambda \left(\lambda_k + \zeta_1(k)(-\beta + \mathbf{1}\{x_T \leq 0\}) \right)
\end{aligned}$$

Unlike the previous actor-critic the parameters update episodically, therefore the algorithm is not fully online.

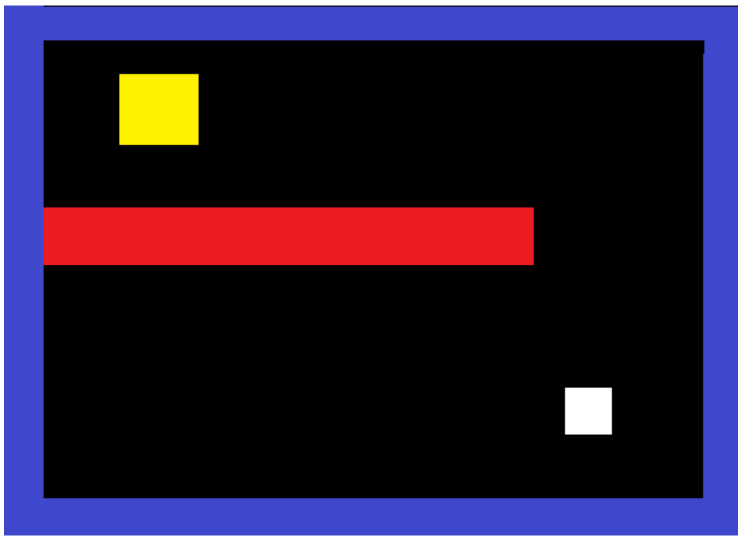
Figure 1: Random Barrier World

4 Experiment

To test the effectiveness of the risk-sensitive MDP approach a world in which a square agent could move around a plane was built using Pygame, the code was adapted from http://programarcadegames.com. The agent was meant to represent a robot moving around a room. In this world a random wall would appear with probability .03, think of this wall as a box or a toy that was sometimes left on the floor. The hypothesis was that agent would learn to be more risk averse using the algorithms proposed in Chow et al. (2015) as opposed to the algorithms described in Sutton and Barto (2017). This experiment differs from the one in Chow et al. (2015) since it is applied to robotics as opposed to finance, and thus a little tweaking of the proposed values was needed. A picture of the world is contained in figure (1). The agent in white received a reward of 500 upon reaching the goal state in gold, -5 for each time step and -50 when it crashed into a wall which are the blue rectangles along the sides. Randomly, the red wall appears which has a reward of -50 like the other walls. A reset was performed each time the agent reached the goal or crashed, as it was frustrating watching a white square continuously hit a blue rectangle. At each time step the agent could either increase his speed up, down, left or right. Note that increasing speed in the opposite direction of the agent's current trajectory would cause him to stop, i.e. if the agent chose to go up and then down it would stop. Also by choosing to increase it's speed up and then left it could move diagonally.

The algorithms compared are the policy gradient with the VaR constraint and the policy gradient algorithm on page 271 of Sutton and Barto (2017) also labeled as REINFORCE. In most engineering applications $\gamma = 1$ and thus the VaR algorithms seemed like a more natural choice. The parameters were initialized as such, $\lambda_0 = 100$ and θ was started with a bit of goal bias. Each parameter was updated after 1000 episodes 10 times for a total of 10,000 episodes. Step sizes were set to $\zeta_1(k) = .01/(i+1) * *0.65$ and $\zeta_2(k) = .01/(i+1) * *0.55$. The weights were optimized such that $\Theta = [-100, 100]^\kappa$ where $\kappa = 124$. After this the resulting policies were run 10,000 times each and the results are contained in figure (2), blue

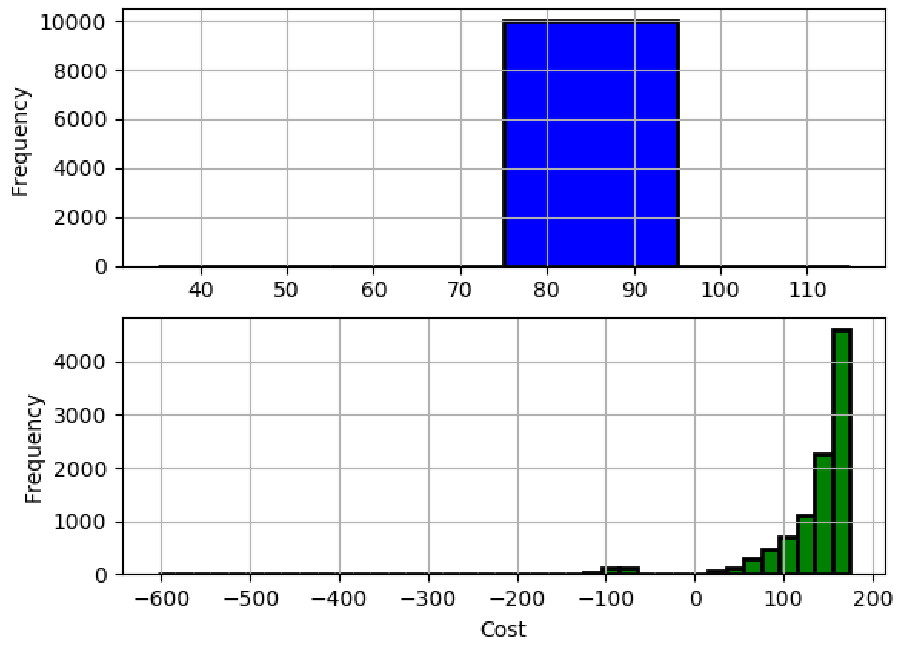

Figure 2: Risk-Sensitive vs Risk-Neutral MDP

Method	Mean	Standard Deviation
Risk-Sensitive	79.98	1.00
Risk Neutral	131.15	59.55

Table 1: Mean and Standard Deviations

for risk-sensitive and green for risk-neutral. The REINFORCE algorithm was adapted to run after a batch of episodes was observed like in the risk-constrained VaR policy gradient algorithm.

5 Conclusion

This semi contrived experiment demonstrated the usefulness of risk-sensitive MDP in a robotics setting. As was hypothesized the risk-neutral algorithm had a larger propensity to hit the wall with a larger standard deviation but a lower mean, as is depicted in figure (2). These results are summarized in table (1). Many episodes, 1000, were run before updating the parameters because the random wall appeared so seldomly that enough episodes were needed for the agent to experience it. The VaR constrained algorithm updated λ by adding a probability to it, $-\beta + \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}\{G_T(j,k) \geq \alpha\}$. When having large rewards and incrementing λ by such a small value causes the updates to have almost no effect. Which is bad because in the θ update λ penalizes bad trajectories which exceeded the predetermined tolerance,

$\lambda_k \sum_{j=1}^{N} \nabla_\theta \log \mathbb{P}_\theta(\xi_{j,k}) \mathbf{1}\{G_T(j,k) \geq \alpha\}$. Thus in the future it would be a smarter idea to scale the rewards.

To Further the ideas of this project it would be interesting to extend the algorithms to continuous action space. This will more accurately depict real world robots. Also it is unrealistic to apply a Monte Carlo type algorithms with a 1000 episodes before updating. In a real robotic setting this would never work. Therefore it would be more interesting to try a more online algorithm like the others described in Chow et al. (2015). Another way to avoid this problem would be to try a model based approach so the agent could already reason about risky events without having to experience them.

References

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

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

Dimitri P Bertsekas. *Nonlinear programming.* Athena scientific Belmont, 1999.

Stefano Carpin, Yin-Lam Chow, and Marco Pavone. Risk aversion in finite markov decision processes using total cost criteria and average value at risk. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 335–342. IEEE, 2016.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *arXiv preprint arXiv:1512.01629*, 2015.

Ronald A Howard and James E Matheson. Risk-sensitive markov decision processes. *Management science*, 18(7):356–369, 1972.

Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.

Harry M Markowitz. *Portfolio selection: efficient diversification of investments*, volume 16. Yale university press, 1968.

LA Prashanth and Mohammad Ghavamzadeh. Actor-critic algorithms for risk-sensitive mdps. In *Advances in neural information processing systems*, pages 252–260, 2013.

R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Yun Shen, Michael J Tobia, Tobias Sommer, and Klaus Obermayer. Risk-sensitive reinforcement learning. *Neural computation*, 26(7):1298–1328, 2014.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 2. MIT press Cambridge, 2017.