Random Barrier Avoidance Using Risk-Constrained MDPs

Lucas Berry

Comp 767

April 20th, 2017

Introduction

In many applications one is concerned with in maximizing expected return while also mitigating risk. This gives rise to the study of risk measures and risk averse policy search. The objects of interest are rare risky events.

Introduction

In many applications one is concerned with in maximizing expected return while also mitigating risk. This gives rise to the study of risk measures and risk averse policy search. The objects of interest are rare risky events.

CVaR and VaR

Two commonly used risk measures are value at risk (VaR) and conditional value at risk (CVaR) also known as expected shorfall (ES) and average value at risk (AVaR). VaR and CVaR are defined as,

$$VaR_{\alpha}(X) := inf\{x|P(X \leq x) \geq \alpha\},$$
$$CVaR_{\alpha}(X) := E\left[X|X \geq VaR_{\alpha}(X)\right].$$

CVaR and VaR

Two commonly used risk measures are value at risk (VaR) and conditional value at risk (CVaR) also known as expected shorfall (ES) and average value at risk (AVaR). VaR and CVaR are defined as,

$$VaR_\alpha(X) := inf\{x|P(X \leq x) \geq \alpha\},$$
$$CVaR_\alpha(X) := E\left[X|X \geq VaR_\alpha(X)\right].$$

CVaR can be rewritten in a more useful way for optimization,

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

where $(x)^+ = \max(0, x)$.

Risk Constrained MDPs

In Chow et. al [2015] the authors derive 5 policy gradient algorithms to solve risk sensitive MDPs constrained by *CVaR* and *VaR*. The optimization problem is rephrased as,

$$\min_{\theta} V^{\theta}(s_0) \qquad \text{subject to} \qquad CVaR_{\alpha}\left(\sum_{k=1}^{T} \gamma^k R_k\right) \leq \beta,$$

Risk Constrained MDPs

In Chow et. al [2015] the authors derive 5 policy gradient algorithms to solve risk sensitive MDPs constrained by *CVaR* and *VaR*. The optimization problem is rephrased as,

$$\min_{\theta} V^{\theta}(s_0) \qquad \text{subject to} \qquad CVaR_{\alpha}\left(\sum_{k=1}^{T} \gamma^k R_k\right) \leq \beta,$$

and

$$\min_{\theta} V^{\theta}(s_0) \qquad \text{subject to} \qquad P\left(\sum_{k=1}^{T} \gamma^k R_k \geq \beta\right) \leq 1 - \alpha.$$

Please note $\alpha \in (0,1)$, $\beta \in \mathbb{R}$ and the constraint in (2) is another way of writing VaR. Also let $G_T = \sum_{k=1}^{T} \gamma^k R_k$.

Finding a Solution

To solve (1) Chow et. al [2015] follows the Lagrangian relaxation procedure described in Chapter 3 of Bertsekas (1999) yielding,

$$\max_{\lambda \geq 0} \min_{\theta, \nu} \left[L(\nu, \theta, \lambda) := V^{\theta}(s_0) + \lambda \left(H_{\alpha} \left(G_T, \nu \right) - \beta \right) \right],$$

where

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

Gradients

Computing the necessary gradients gives

$$\nabla_\theta L(\nu,\theta,\lambda) = \nabla_\theta V^\theta(s_0) + \frac{\lambda}{1-\alpha}\nabla_\theta E\left[(G_T-\nu)^+\right],$$
$$\partial_\nu L(\nu,\theta,\lambda) = \lambda\left(1 + \frac{1}{1-\alpha}\partial_\nu E\left[(G_T-\nu)^+\right]\right),$$
$$\nabla_\lambda L(\nu,\theta,\lambda) = \nu + \frac{1}{1-\alpha}E\left[(G_T-\nu)^+\right].$$

Monte-Carlo

ν **Update**:

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

θ **Update**:

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

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

Monte-Carlo

λ **Update**:

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

Random Barriers

In Chow et. al [2015] they used their algorithm for a financial mathematics problem. I wanted to apply it to a robotics problem.