Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Eidgenössische Technische Hochschule Zü
Swiss Federal Institute of Technology Zuric

Prof. Name

Master Thesis

LaTeX -Latex Template

Spring Term 2020

Supervised by:
Supervisor 1

Author:
Nuria Armengol Urpi

Contents

Preface

Bla bla ...

Abstract

Bla bla ...

Symbols

Symbols

ϕ, θ, ψ	roll, pitch and yaw angle

Indices

x	x axis
y	y axis

Acronyms and Abbreviations

ETH	Eidgenössische Technische Hochschule

Chapter 1

Introduction

Introduction

1.1 Distributional RL

Recent research has been done demonstrating the importance of learning the value distribution, i.e, the distribution of the random return received by a RL agent. This differs from the common RL approach which is focused on learning the expected value of this return.
One of the major goals of RL is to teach an agent so that it learns how to act so that it maximizes its expected utility, Q [**?**] Bellman's equation describes this value Q in terms of the expected reward and expected outcome of the random transition $(x, a) \to (X', A')$, showing the particular recursive relationship between the value of a state and the values of its successor states:

$$Q(x, a) = \mathbb{E}[R(x, a)] + \gamma \mathbb{E}[Q(X', A')] \tag{1.1}$$

Distributional RL aims to go beyond the notion of *value* and training to study instead the random return Z.

1.1.1 Example showing interest in learning the distribution

Imagine the example in which we are playing a board game and we roll 2 dices. If we get a 3, we fall in prison and need to pay 2000CHF (ie reward of -2000CHF), whereas otherwise we we collect a salary of 200CHF (ie reward of +200CHF). If we consider the common reinforcement learning approach and we compute the expected immediate ($\gamma = 1$) reward:

$$\mathbb{E}[R(x)] = \frac{1}{36}(-2000\,\text{CHF}) + \frac{35}{36}(200\,\text{CHF}) = 138.88\,\text{CHF} \tag{1.2}$$

Hence, the expected immediate return is +138.88CHF. However, in any case we will get a return of +138.88CHF. Instead:

$$R(x) = \begin{cases} -2000\,\text{CHF}, & \text{w.p } \frac{1}{36} \\ 200\,\text{CHF}, & \text{w.p } \frac{35}{36} \end{cases}$$

We define the random return $Z^\pi(x,a)$ as the random variable that represents the sum of discounted rewards obtained by starting from position x taking action a and thereupon following policy π.
This variable captures intrinsic randomness from:

1. Immediate rewards
2. Stochastic dynamics
3. Possibly an stochastic policy

Having defined $Z^\pi(x,a)$, we can clearly see that:

$$Q^\pi(x,a) = \mathbb{E}[Z^\pi(x,a)] \tag{1.3}$$

Z is also described by a recursive equation, but of a distributional nature:

$$Z^\pi(x,a) \overset{D}{=} R(x,a) + \gamma Z(x',a') \tag{1.4}$$

where $x' \backsim\ p(\cdot|x,a)$ and $a' \backsim\ \pi(\cdot|x')$
where $\overset{D}{=}$ denotes that the RV on both sides of the equation share the same probability distribution. The *distributional Bellman equation* defined in (1.4), states that the distribution of Z is characterized by the interaction of 3 RV's: the random variable reward R, the next state-action (X',A') and its random return Z(X',A'). From here on, we will view Z^π as a mapping from state-action pairs to distributions over returns, and we call this distribution the *value distribution.*

1.1.2 Distributional Bellman Operator

In the policy evaluation setting [**?**], one aims to find the value function $V\pi$ associated with a given fixed policy π. In the distributional case, we aim to find $Z\pi$. [**?**] defined the Distributional Bellman operator T^π. We view the reward function as a random vector $\mathrm{R} \in \mathbb{Z}$ and define the transition operator $P^\pi : \mathbb{Z} \to \mathbb{Z}$

$$P^\pi Z(x,a) \overset{D}{=} Z(X',A') \tag{1.5}$$

$$X' \backsim\ P(\cdot|x,a) \text{ and } A' \backsim\ \pi(\cdot|X') \tag{1.6}$$

where we use capital letters to emphasize the random nature of the next state-action pair (X',A') Then, the Distributional Bellman operator T^π is defined as:

$$T^\pi Z(x,a) \overset{D}{=} R(x,a) + \gamma P^\pi Z(x,a) \tag{1.7}$$

[**?**] showed that (1.7) is a contraction mapping whose unique fixed point is the random return Z^π.

Wasserstein metric:

The p-Wasserstein metric W_p, for $p \in [1,\infty]$, also known as the Earth Mover's Distance when $p = 1$ is an integral probability metric between distributions. The p-Wasserstein distance is characterized as the L^p metric on inverse cumulative distribution functions (CDF). Tht is, the p-Wasserstein metric between distributions U and Y is given by:

$$W_p(U,Y) = \Big(\int_0^1 |F_Y^{-1}(w) - F_U^{-1}(w)|^p dw \Big)^{\frac{1}{p}} \tag{1.8}$$

where for a random variable Y, the inverse CDF F_Y^{-1} of Y is defined by:

$$F_Y^{-1}(w) \coloneqq \inf\{y \in \mathbb{R} \mid w \leq F_Y(w)\} \tag{1.9}$$

where $F_Y(w) = Pr(y \leq Y)$.

Add Figure 2.1 in [?]

Unlike the Kullback-Leibler divergence , the Wasserstein metric is a true probability metric and considers both the probability of and the distance between various outcome events, which makes it well-suited to domains where an underlying similarity in outcome is more important than exactly matching likelihoods.

Contraction in $\hat{d}_p$:

Let $\mathcal{Z}$ be the space of action-value distributions:

$$\mathcal{Z} = \{Z \mid \mathcal{X} \times \mathcal{A} \to \wp(\mathbb{R}) \tag{1.10}$$

$$\mathbb{E}[|Z(x,a)|^p < \infty, \forall (x,a), p \geq 1]\} \tag{1.11}$$

check first line the $\wp$

Then, for two action-value distribution $Z_1, Z_2 \in \mathcal{Z}$, the maximal form of the Wasserstein metric is defined by:

$$\hat{d}_p(Z_1, Z_2) \coloneqq \sup_{x,a} W_p(Z_1(x,a), Z_2(x,a)) \tag{1.12}$$

[?] showed that $\hat{d}_p$ is a metric over value distributions and furthermore, the distributional Bellman operator T^π is a contraction in $\hat{d}_p$. Consider the process $Z_{k+1} \coloneqq T^\pi Z_k$, starting with some $Z_0 \in \mathcal{Z}$.
$T^\pi Z : \mathcal{Z} \to \mathcal{Z}$ is a γ-contraction in the Wasserstein metric $\hat{d}_p$, which implies that not only the first moment (expectation) converges exponentially to Q^π, but also in all moments.
Lemma 1: (Lemma 3 in [?])
T^π is a γ-contraction: for any two $Z_1, Z_2 \in \mathcal{Z}$,

$$\hat{d}_p(T^\pi Z_1, T^\pi Z_2) \leq \gamma \hat{d}_p(Z_1, Z_2) \tag{1.13}$$

Using Banach's fixed point theorem, it is proven that T^π has a unique fixed point, which by inspection must be Z^π.
Hence the $\hat{d}_p$ metric is shown to be useful metric for studying behavior of distributional RL algorithms, and to showed their convergence to a fixed point. Moreover, shows than en effective way to learn a value distribution is to attempt minimize the Wasserstein distance between a distribution Z and its distributional Bellman update $T^\pi Z$, analogously to the way that TD-learning attempts to iteratively minimize the L^2 distance between Q and TQ.
We have so fare considered a policy evaluation setting, ie trying to learn a value distribution for a fixed policy π, and we studied the behavior of its associated distributional operator T^π. In the control setting, ie, when we try to find a policy π^* that maximizes a value, or its distributional analogous, ie that induces an optimal value distribution. However, while all optimal policies attain the same value Q^*, in general there are many optimal value distributions.
The distributional analogue of the Bellman optimality operator converges, in a weak sense, to the set of optimal value distributions, but this operator is *not a contraction in any metric between distributions.*.
Let Π^* be the set of optimal policies.
Definition 1: An optimal value distribution is the value distribution of an optimal policy. The set of optimal value distributions is

$$\mathcal{Z}^* \coloneqq \{Z^{\pi^*} | \pi^* \in \Pi^*\}$$

Not all value distributions with expectation Q^* are optimal, but they must match the full distribution of the return under some optimal policy. **Definition 2:** A greedy policy π for Z $in\mathcal{Z}$ maximizes the expectation of Z. The set of greedy policies for Z is:

$$\mathcal{G}_{\mathcal{Z}} \coloneqq \{\pi \mid \sum_a \pi(a|x)\mathbb{E}(Z(x,a) = \max_{a' \in \mathcal{A}} Q(x',a')\}$$

We will call a *distributional Bellman optimality operator* any operator $\mathcal{T}$ which implements a greedy selection rule, ie:

$$\mathcal{T}Z = \{\mathcal{T}^{\approx} Z \text{ for some } \pi \in \mathcal{G}_{\mathcal{Z}}\}$$

As in the policy evaluation setting, we are interested in the behavior of the iterates $Z_{k+1} := \mathcal{T}Z_k, Z_O \in \mathcal{Z}$. Lemma 4 in [**?**] shows that $\mathbb{E}Z_k$ behaves as expected: **Lemma 4:** Let $Z_1, Z_2 \in \mathcal{Z}$. Then:

$$\|\mathbb{E}\mathcal{T}Z_1 - \mathbb{E}\mathcal{T}Z_2\|_\infty \leq \gamma \|\mathbb{E}Z_1 - \mathbb{E}Z_2\|_\infty$$

and in particular $\mathbb{E}Z_1 \to Q^*$ exponentially quickly. However, Z_k is not assured to converge to a fixed point. Specifically, they provide a number of negative results concerning $\mathcal{T}$:
Proposition 1: The operator $\mathcal{T}$ is not a contraction.
Proposition 2: Not all optimality operators have a fixed point $Z^* = \mathcal{T}Z^*$ **Proposition 3**: That $\mathcal{T}$ has a fixed point $Z^* = \mathcal{T}Z^*$ is insufficient to guarantee the convergence of $\{Z_k\}$ to Z^*
Another result, shows that we cannot in general minimize the Wasserstein metric, viewed as a loss, using stochastic gradient descent methods. This limitation, is crucial in a practical context, when the value distribution needs to be approximated.

1.1.3 Quantile approximation

[**?**] used the theory of quantile regression [**?**], to design an algorithm applicable in a stochastic approximation setting. Quantile regression is used to estimate the quantile function at precisely chosen points. Then the Bellman update is applied onto this parameterized quantile distribution. This combined operator is proven to be a contraction and the estimated quantile function is shown to converge to the true value distribution when minimized using stochastic approximation.

1.1.4 Quantile projection:

Our current aim is to estimate quantiles of the target distribution, ie the values of the return that divide the value distribution in equally sized parts. We will call it a quantile distribution, and we will let $\mathcal{Z}_{\mathcal{Q}}$ be the space of quantile distributions. We denote the cumulative probabilities associated with such a distribution by $\tau_1, \tau_2 ... \tau_N$, so that $\tau_i = \frac{i}{N}$ for $i = 1, ...N$.
Formally, let $\theta : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^N$ be some parametric model. A quantile distribution $Z_\theta \in \mathcal{Z}_{\mathcal{Q}}$ maps each state-action pair (x,a) to a uniform probability distribution supported on $\{\theta_i(x,a)\}$. Hence we can approximate it by a uniform mixture of N Diracs:

$$Z_\theta(x,a) \coloneqq \frac{1}{N} \sum_{i=1}^{N} \delta_{\theta_i(x,a)} \tag{1.14}$$

with each θ_i assigned a fixed quantile. We aim to learn the support of these Diracs, i.e learn $\theta_i \forall i, a, x$. We will do it by quantifying the projection of an arbitrary value distribution $Z \in \mathcal{Z}$ onto $\mathcal{Z}_{\mathcal{Q}}$, that is:

$$\prod_{W_1} Z \coloneqq \underset{Z_\theta \in \mathcal{Z}_\mathcal{Q}}{\arg\min} W_1(Z, Z_\theta) \tag{1.15}$$

This projection $\prod_{W_1}$ is the quantile projection.

get the W1 next to the prod not below

We can quantify the projection between a distribution with bounded first moment Y and U, a uniform distribution over N Diracs as in (1.15) with support $\{\theta_1, ...\theta_N\}$ by:

$$W_1(Y, U) = \sum_{i=1}^{N} \int_{\tau_{i-1}}^{\tau_i} |F_Y^{-1}(w) - \theta_i| dw \tag{1.16}$$

Lemma 2 in [**?**] establishes that the values $\{\theta_1, ...\theta_N\}$ for the returns that minimize $W_Y(Y, U)$ are given by $\theta_i = F_Y^- 1(\hat{\tau}_i)$, where $\hat{\tau}_i = \frac{\tau_{i-1}+\tau_i}{2}$.

1.1.5 Quantile Regression

Quantile regression is a method for approximating quantile functions of distributions. The quantile regression loss, for quantile $\tau \in [0, 1]$, is an asymmetric convex lox function that penalizes underestimation errors with weight τ and overestimation errors with weight $1 - \tau$.
For a distribution Z, and given quantile τ, the value of the quantile function $F_Z^{-1}(\tau)$ may be characterized as the minimizer of the quantile regression loss:

$$\mathcal{L}_{QR}^{\tau}(\theta) = \mathbb{E}_{\hat{Z} \backsim Z}[\rho_\tau(\hat{Z} - \theta)] \tag{1.17}$$

$$\rho_\tau(u) = u(\tau - \delta_{u<0}), \forall u \in \mathbb{R} \tag{1.18}$$

Given that the minimizer of the quantile regression loss for τ is $F_Z^{-1}(\tau)$, and using Lemma 2 in [**?**], which claims that the values of $\{\theta_1, ...\theta_N\}$ that minimize $W_1(Z, Z_\theta)$ are given by $\theta_i = F_Y^- 1(\hat{\tau}_i)$; we can claim that the values of $\{\theta_1, ...\theta_N\}$ are the minimizers of the following objective:

$$\sum_{i=1}^{N} \mathbb{E}_{\hat{Z} \sim Z}[\rho_{\hat{\tau}_i}(\hat{Z} - \theta_i)] \tag{1.19}$$

This loss gives unbiased sample gradients and hence, we can find the minimizing $\{\theta_1, ...\theta_N\}$ by stochastic gradient descent.

add huberloss

Proposition 2 in [**?**] states that the combined quantile projection $\prod_{W_1}$ with the Bellman update $\mathcal{T}^\pi$ has a unique fixed point $\hat{Z}^\pi$, and the repeated application of this operator, or its stochastic approximation, converges to $\hat{Z}^\pi$.

1.1.6 Quantile Regression Temporal Difference Learning

Temporal difference learning updates the estimated value function with a single unbiased saple following policy π. Quantile regression allows to improve the estimate of the quantile function for some target distribution Y(x), by observing samples $y \sim Y(x)$ and minimizing equation (1.18). Using the quantile regression loss, we can obtain an approximation with minimal 1-Wasserstein distance from the original. We can combine this with the distributional Bellman operator to give a target distribution for quantile regression, creating the quantile regression temporal differnce learning algorithm:

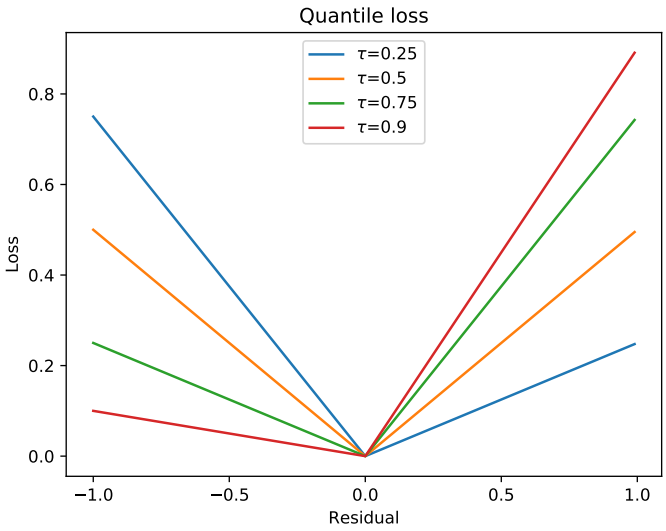

Figure 1.1: Car trajectory using DDPG algorithm and velocity penalization with probability P=0.2

$$u = r + \gamma z' - \theta_i(x) \tag{1.20}$$

$$\theta_i(x) \leftarrow \theta_i(x) + \alpha(\hat{\tau}_i - \delta_{u<0}) \tag{1.21}$$

$$a \backsim \pi(\cdot|x), r \backsim R(x,a), x' \backsim P(\cdot|x,a), z' \backsim Z_\theta(x') \tag{1.22}$$

where Z_θ is a quantile distribution as in (1.15) and $\theta_i(x)$ is the estimated value of $F^{-1}_{Z^\pi(x)}(\hat{\tau}_i)$ in state x.

1.2 IDEA AND CAR EXPERIMENT

We focus on the importance of *value distribution*, the distribution of the random return received by a RL agent, in contrast to the common approach in RL of modelling the expectation of this return. The latter neither takes int account the variability of the cost (i.e fluctuations around the mean), nor its sensitivity to modeling errors. [**?**]
We aim to learn this distribution and try to minimize other metrics rather than its mean, which can be crucial for some environments in which ensuring that the cost is always above a certain value with certain probability is crucial.
A metric that has recently gained a lot of popularity is the Conditional Value at Risk, eg in finance, due to its favorable computation properties and superior ability to safeguard a decision maker from the "outcomes that hurt the most" [**?**]

1.2.1 Conditional Value-at-Risk (CVaR)

Let Z be a bounded-mean random variable, i.e $\mathbb{E}[|Z|] < \infty$, on a probability space $(\Omega, \mathbb{F}, \mathbb{P})$, with cumulative distribution function $F(z) = \mathbb{P}(Z \leq z)$. We interpret Z as a reward. The value-at-risk (VaR) at confidence level $\alpha \in (0, 1)$ is the α quantile of Z, i.e, $\mathrm{VaR}_\alpha(Z) = \inf\{z \mid F(z) \geq \alpha\}$. The conditional value-at-risk (CVaR) at confidence level $\alpha \in (0, 1)$ is defined as the expected reward of outcomes worse than the α-quantile (VaR_α):

$$\mathrm{CVaR}_\alpha(Z) = \frac{1}{\alpha}\int_0^\alpha F_Z^{-1}(\beta)d\beta = \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\beta(Z)d\beta \tag{1.23}$$

Rockafellar and Uryasev [**?**] also showed that CVaR is equivalent to the solution of the following optimization problem:

$$\mathrm{CVaR}_\alpha(Z) = \min_\nu\big\{\nu + \frac{1}{\alpha}\mathbb{E}_Z[[Z-\nu]^-]\big\} \tag{1.24}$$

where $(x)^- = \min(x, 0)$. In the optimal point it holds that $\nu^* = \mathrm{VaR}_\alpha(Z)$.
A useful property of CVaR, is its alternative dual representation [**?**]:

$$\mathrm{CVaR}_\alpha(Z) = \min_{\xi \in U_{\mathrm{CVaR}}(\alpha,\mathbb{P})} \mathbb{E}_\xi[Z] \tag{1.25}$$

where $\mathbb{E}_\xi[Z]$ denotes the ξ-weighted expectation of Z, and the risk envelope U_{CVaR} is given by:

$$U_{\mathrm{CVaR}}(\alpha, \mathbb{P}) = \Big\{\xi | \xi(w) \in \big[0, \frac{1}{\alpha}\int_{w\in\Omega} \xi(w)\mathbb{P}(w)dw = 1\big]\Big\} \tag{1.26}$$

Thus, the CVaR of a random variable may be interpreted as the worst case expectation of Z, under a perturbed distribution $\xi\mathbb{P}$.

1.3 Current results

Problem: A car with fully-observable 2D-state: [position, velocity] needs to move from initial position $x_0 = 0$m and initial velocity $v_0 = 0\text{m}\,ø^{-1}$ to goal position $x_F = 2.0\,\text{m}$. At every time-step ø, with a discretization of $t_d = 0.1$, the car receives an acceleration as a control input. The control input a is constrained to range between [-1.0,1.0]$\text{m}\,ø^{-2}$. Per every time-step passed before it reaches the goal, it receives a penalization reward $R_\tau = -1$ If the car reaches the goal position, it receives a reward $R_F = 100$ and the episode ends. Otherwise, after $T_F = 400ø$ the episode ends (with no extra penalization).

1.3.1 Case 1: No velocity penalization

Both DDPG and CVAR-DDPG manage to arrive with linear increment in velocity with a slope of 2. (Accelearation at maximum during the whole episode). For this setup we have:

$$x = x_0 + v_0\frac{ø}{10} + 0.5a\left(\frac{ø}{10}\right)^2$$

In the optimal case, the car keeps an acceleration of $1\text{m}\,ø^{-2}$ for the whole episode, and hence reaches $x_F = 2$m with 20 time-steps. Hence the final cumulative reward $G_T = (20+1)R_\tau + R_F = 70$. **add pics**

add pics

1.3.2 Case 2: Velocity penalization with probability 1

The experiment is carried out to ensure the two algorithms manage to learn the new reward function when there is no uncertainty.

In this setup, when the car velocity exceeds $1\text{m}\,ø^{-1}$, it receives a penalization of $R_v = -20$. We expect both algorithms to perform similarly since there is no reward uncertainty.
As expected, both algorithms learn to accelerate with maximum value till a velocity of $1\text{m}\,ø^{-1}$ is reached, and then keeping this velocity constant until the goal is reached.

Starting from $x_0 = 0$m, the car reaches a velocity of $1\text{m}\,ø^{-1}$ after 10 time-steps, at $x_{\tau=10} = 0.5$. Keeping velocity $1\text{m}\,ø^{-1}$ through the rest of the episode, it reaches the goal position after 14 time-steps. Hence the final cumulative reward $G_T = (10+14+1)R_\tau + R_F = 74$. The reward values were chosen in order to make sure that, for this Case 2 setting, driving with a velocity higher than $1\text{m}\,ø^{-1}$ never induces higher cumulative rewards.

For this Case 2 setting, the trained models were saved using Early stopping with *maximal reward in episode evaluation* as a metric and with a patience of 100 episodes.
The quantiles used for learning the actor for the CVAR-DDPG algorithm were sampled uniformly $\backsim\ U[0,1]$

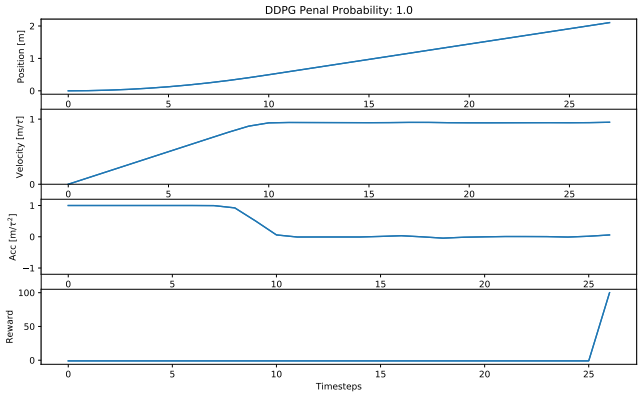

Figure 1.2: Car trajectory using DDPG algorithm and velocity penalization with probability 1

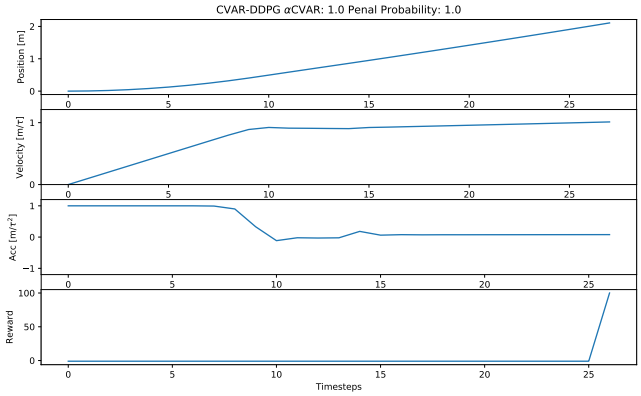

Figure 1.3: Car trajectory using CVAR-DDPG algorithm and velocity penalization with probability 1. (α-CVAR = 1)

1.3.3 Case 3: Velocity penalization with probability P

The experiment is carried out to show the risk-sensitiveness property of the CVAR-DDPG algorithm.

The models saved were the ones that obtained a maximum CVAR (with a window of 10 episodes) of the cumulative rewards during evaluation
The quantiles used for learning the actor for the CVAR-DDPG algorithm were sampled uniformly $\backsim\ U[0, \alpha]$ where $\alpha = 0.2$

For P=0.2 the CVAR-DDPG algorithm learns to saturate the velocity, even though the probability of a penalization is low, whereas the DDPG algorithm doesn't, and keeps a linear increase of the velocity during the whole episode. The CVAR algorithm reaches its maximum CVAR of 64.0 at episode 220, whereas the DDPG reaches its maximum CVAR value of 49.0 at episode 1435.

Important issue: Although CVAR-DDPG finds a risk-sensitive trajectory at

episode 220, it doesn't converge there and keeps oscillating and even moves towards a risk-neutral behaviour later on.
The graph in figure 1.8 , shows the evolution of the sampled mean of the tail of the sampled cumulative value distribution (CDF). (ie we compute via IQN the quantile values from the tail value distribution (VD) and take the mean). The value it converges to coincides with the maximum value of the CVAR we achieved ,but then the actor doesn't seem to behave accordingly.
However I am now thinking maybe computing the mean of the tail of the **CDF** and trying to push the actor to maximize it, it is not the best approach since, it is not the mean of the tail of the **CDF** what we aim to, but the mean of the tail of the VD. What do you think?

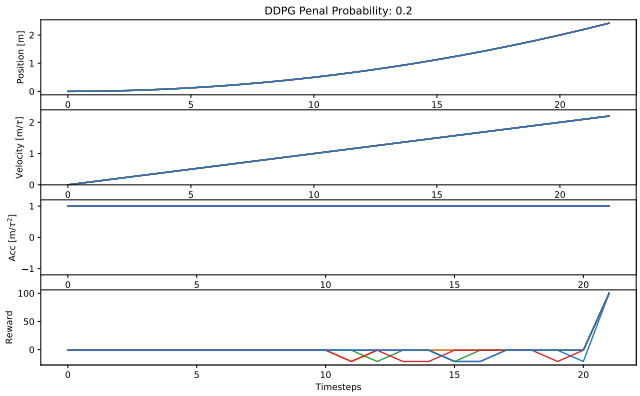

Figure 1.4: Car trajectory using DDPG algorithm and velocity penalization with probability P=0.2

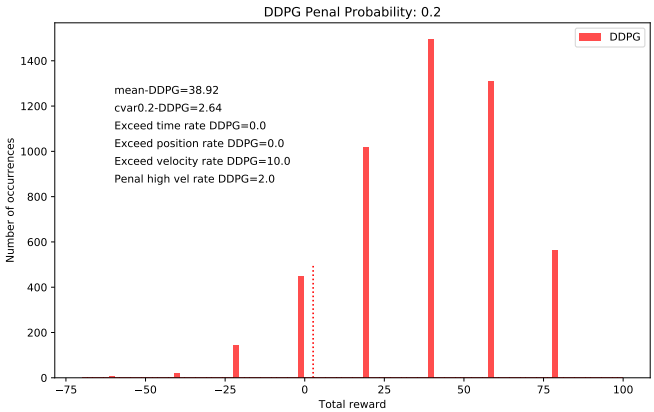

Figure 1.5: Reward distribution using DDPG algorithm and velocity penalization with probability 0.2

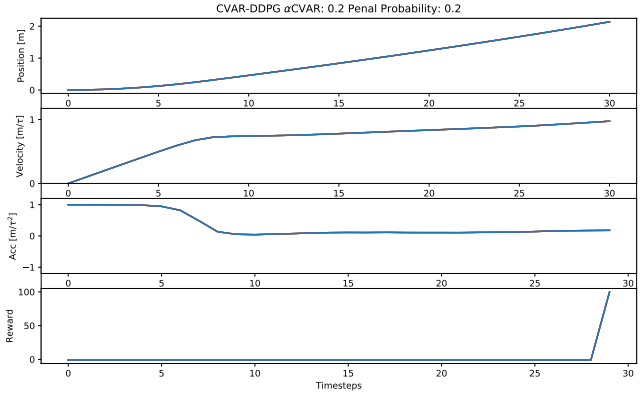

Figure 1.6: Car trajectory using CVAR-DDPG algorithm and velocity penalization with probability 0.2 and (α-CVAR = 0.2)

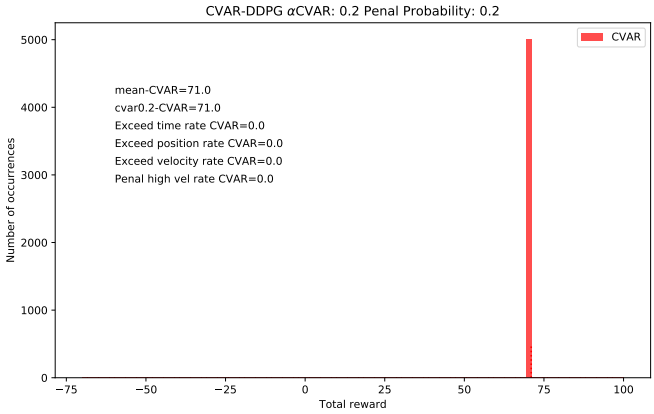

Figure 1.7: Reward distribution using CVAR-DDPG algorithm and velocity penalization with probability 0.2 and (α-CVAR = 0.2)

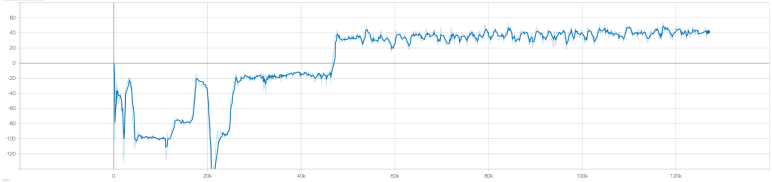

Figure 1.8: Evolution of the sampled mean of the tail of the Cumulative Value Distribution during training epochs