Risk-Averse Trust Region Optimization for Reward-Volatility Reduction

Lorenzo Bisi^{1,2*†} , **Luca Sabbioni**^{1,2†} , **Edoardo Vittori**^{1,3†} , **Matteo Papini**^{1} and **Marcello Restelli**^{1}
^{1}Politecnico di Milano
^{2}ISI Foundation
^{3}Banca IMI
{lorenzo.bisi, luca.sabbioni, edoardo.vittori, matteo.papini, marcello.restelli}@polimi.it,

Abstract

The use of reinforcement learning in algorithmic trading is of growing interest, since it offers the opportunity of making profit through the development of autonomous artificial traders, that do not depend on hard-coded rules. In such a framework, keeping uncertainty under control is as important as maximizing expected returns. Risk aversion has been addressed in reinforcement learning through measures related to the distribution of returns. However, in trading it is essential to keep under control the risk of portfolio positions in the intermediate steps. In this paper, we define a novel measure of risk, which we call reward volatility, consisting of the variance of the rewards under the state-occupancy measure. This new risk measure is shown to bound the return variance so that reducing the former also constrains the latter. We derive a policy gradient theorem with a new objective function that exploits the mean-volatility relationship. Furthermore, we adapt TRPO, the well-known policy gradient algorithm with monotonic improvement guarantees, in a risk-averse manner. Finally, we test the proposed approach in two financial environments using real market data.

1 Introduction

Reinforcement Learning (RL) [Sutton and Barto, 1998] methods have a quite important recent history in financial trading, starting with [Moody and Saffell, 2001] and followed by several others such as [Shen *et al.*, 2014]. However, even if interesting from a theoretical point of view, these approaches are still not mature for real applications, due to scalability or slow convergence issues. Recently, policy search [Deisenroth *et al.*, 2013] algorithms, such as TRPO [Schulman *et al.*, 2015] and PPO [Schulman *et al.*, 2017], have achieved great results [OpenAI, 2018 accessed May 2020; Heess *et al.*, 2017] in terms of efficiently maximizing the expected value of the cumulative discounted rewards (referred to as *expected return*). Nonetheless, while proving very effective when the objective is the sole maximization of the return (even in the case of partially observable, non-Markovian environments), they are not ideal in the trading framework where keeping a low risk is mandatory. The focus of this research is to develop a RL algorithm capable of balancing risk and return, while taking advantage of the improved performance of current state-of-the-art algorithms.

Risk-aversion in reinforcement learning has been taken into account with many different approaches [García and Fernández, 2015]: employing a utility function for the return [Shen *et al.*, 2014], changing the objective function, or adding a constraint [Di Castro *et al.*, 2012]. A number of modified objectives have been studied, for example the minimization of variance of the returns (referred to as return variance throughout the paper), in a mean-variance [Tamar and Mannor, 2013; Prashanth and Ghavamzadeh, 2014b] or Sharpe ratio [Moody and Saffell, 2001] fashion. Another example is a family of well-behaved risk measures, which includes CVaR, called coherent risk measures [Tamar *et al.*, 2017]. Nevertheless, all these approaches consider only the minimization of the long-term risk, while in financial trading interim results are also fundamental, and keeping a low-varying intermediate P&L (Profit and Loss) becomes crucial. This paper formally defines and analyzes for the first time, to the best of our knowledge, the variance of the reward at each time step w.r.t. state visitation probabilities. We call this quantity *reward volatility*. Intuitively, the return variance measures the variation of cumulated rewards among trajectories, while reward volatility is concerned with the variation of single-step rewards among visited states. We derive a Bellman equation for the reward-volatility that is exploited to obtain a policy gradient theorem for this novel objective. In addition, we also show that this new measure upper bounds the return variance (albeit for a normalization term). This is an interesting outcome, indicating that it is possible to use the analytic results we derived for the reward volatility to keep under control the return variance. Reward volatility is used to define a new risk-averse performance objective, called *mean-volatility*, which is a trade-off between the maximization of the expected return and the minimization of short-term risk. This trade-off can be customized in order to meet the specific needs of each individual trader, by tuning the risk aversion parameter. Optimizing the mean-volatility objective allows to limit the *inherent risk* due to the stochastic nature of the environment. How-

[*] Contact author
[†] Equal contribution

ever, the imperfect knowledge of the model parameters, and the consequent imprecise optimization process, is another relevant source of risk, known as *model risk*. This is especially important when the optimization is performed on-line, as may happen for an autonomous, adaptive trading system. To avoid any kind of performance oscillation, the intermediate solutions implemented by the learning algorithm must guarantee continuing improvement. The TRPO algorithm [Schulman *et al.*, 2015] provides this kind of guarantees (at least in its ideal formulation) for the risk-neutral objective, based on the conservative bounds proven in [Kakade and Langford, 2002]. Thanks to the linearity of the corresponding Bellman equation, we can show that the same bound still holds under the mean-volatility formulation. Hence, we derive the Trust Region Volatility Optimization (TRVO) algorithm, a TRPO-style algorithm for the new mean-volatility objective.

This paper is organized as follows: after some background on MDPs and on policy gradients (Section 2), the volatility measure is introduced in Section 3 and compared to the return variance. The Policy Gradient Theorem for the mean-volatility objective is provided in Section 4. In Section 4.1, we introduce an estimator for the gradient which is based on sample trajectories obtained from direct interaction with the environment. In Section 5, the monotonic improvement guarantees are presented and discussed, and the TRVO algorithm is introduced. Finally, in Section 7, we test our algorithms on two financial environments, where the agents must learn to trade on real assets using historical data.

2 Preliminaries

A discrete-time Markov Decision Process (MDP) is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mu \rangle$, where $\mathcal{S}$ is the (continuous) state space, $\mathcal{A}$ the (continuous) action space, $\mathcal{P}(\cdot|s,a)$ is a Markovian transition model that assigns to each state-action pair (s,a) the probability of reaching the next state s', $\mathcal{R}$ is a bounded reward function, i.e. $\sup_{s\in\mathcal{S},a\in\mathcal{A}} |\mathcal{R}(s,a)| \leq R_{\max}$, $\gamma \in [0,1)$ is the discount factor, and μ is the initial state distribution. The policy of an agent is characterized by $\pi(\cdot|s)$, which assigns to each state s the density distribution over the action space $\mathcal{A}$.

We consider infinite-horizon problems in which future rewards are exponentially discounted with γ. Following a trajectory $\tau := (s_0, a_0, s_1, a_1, s_2, a_2, ...)$, let the returns be defined as the discounted cumulative reward: $G_\tau = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)$. For each state s and action a, the action-value function is defined as:

$$Q_\pi(s,a) := \mathop{\mathbb{E}}_{\substack{s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)\\ a_{t+1}\sim\pi(\cdot|s_{t+1})}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t,a_t) | s_0 = s, a_0 = a\right], \tag{1}$$

which can be recursively defined by the following Bellman equation:

$$Q_\pi(s,a) = \mathcal{R}(s,a) + \gamma \mathop{\mathbb{E}}_{\substack{s'\sim\mathcal{P}(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}} \left[Q_\pi(s',a')\right].$$

For each state s, we define the state-value function of the stationary policy $\pi(\cdot|s)$ as:

$$V_\pi(s) := \mathop{\mathbb{E}}_{\substack{a_t\sim\pi(\cdot|s_t)\\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t,a_t) | s_0 = s\right] \tag{2}$$

It is useful to introduce the (discounted) state-occupancy measure induced by π:

$$d_{\mu,\pi}(s) := (1-\gamma) \int_S \mu(s_0) \sum_{t=0}^{\infty} \gamma^t p_\pi(s_0 \xrightarrow{t} s) \, \mathrm{d}s_0,$$

where $p_\pi(s_0 \xrightarrow{t} s)$ is the probability of reaching state s in t steps from s_0 following policy π. The objective is the *normalized*[1] expected return: J_π. It is defined below using two distinct formulations, one based on transition probabilities and the other on the state occupancy $d_{\mu,\pi}$:

$$\begin{aligned} J_\pi &:= (1-\gamma) \mathop{\mathbb{E}}_{\substack{s_0\sim\mu\\ a_t\sim\pi(\cdot|s_t)\\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t,a_t)\right] \\ &= \mathop{\mathbb{E}}_{\substack{s\sim d_{\mu,\pi}\\ a\sim\pi(\cdot|s)}} \left[\mathcal{R}(s,a)\right]. \end{aligned}$$

For the rest of the paper, we consider parametric policies, where the policy $\pi_{\boldsymbol{\theta}}$ is parametrized by a vector $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^m$.[2]

3 Risk Measures

This section introduces the concept of reward volatility, comparing it with the more common return variance. The latter, denoted with σ^2_π, is defined as:

$$\sigma^2_\pi := \mathop{\mathbb{E}}_{\substack{s_0\sim\mu\\ a_t\sim\pi_{\boldsymbol{\theta}}(\cdot|s_t)\\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)}} \left[\left(\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t,a_t) - \frac{J_\pi}{1-\gamma}\right)^2\right]. \tag{3}$$

In our case, it is useful to define *reward volatility* ν^2_π in terms of the distribution $d_{\mu,\pi}$. As it is not possible to define the return variance in the same way, we also rewrite reward volatility as an expected sum over trajectories:[3]

$$\nu^2_\pi := \mathop{\mathbb{E}}_{\substack{s\sim d_{\mu,\pi}\\ a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}} \left[(\mathcal{R}(s,a) - J_\pi)^2\right] \tag{4}$$

$$= (1-\gamma) \mathop{\mathbb{E}}_{\substack{s_0\sim\mu\\ a_t\sim\pi_{\boldsymbol{\theta}}(\cdot|s_t)\\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \left(\mathcal{R}(s_t,a_t) - J_\pi\right)^2\right]. \tag{5}$$

[1] In our notation, the expected return (as commonly defined in the RL literature) is $J_\pi/(1-\gamma)$.

[2] For the sake of brevity, when a variable depends on the policy $\pi_{\boldsymbol{\theta}}$, in subscripts only π is shown, omitting the dependency on $\boldsymbol{\theta}$.

[3] In finance, the term "volatility" refers to a generic measure of variation, often defined as a standard deviation. In this paper, volatility is defined as a variance.

Once we have set a mean-variance parameter λ, the *performance* or objective function related to the policy π can be defined as:

$$\eta_\pi := J_\pi - \lambda \nu_\pi^2, \tag{6}$$

called *mean-volatility* hereafter, where $\lambda \geq 0$ allows to trade-off expected return maximization with risk minimization. Similarly, the mean-variance objective is $J_\pi/(1-\gamma) - \lambda\sigma_\pi^2$. An important result on the relationship between the two variance measures is the following:

Lemma 1 *Consider the return variance σ_π^2 defined in Equation* (3) *and the reward volatility ν_π^2 defined in Equation* (5). *The following inequality holds:*

$$\sigma_\pi^2 \leq \frac{\nu_\pi^2}{(1-\gamma)^2},$$

Sketch of Proof. Expanding the square term in Equation (3), $\sigma_\pi^2 = \mathbb{E}_\tau\big[(\sum_t \gamma^t R_t)^2\big] - J_\pi^2/(1-\gamma)^2$. As a consequence of the Cauchy-Schwarz inequality, $\mathbb{E}_\tau\big[(\sum_t \gamma^t R_t)^2\big] \leq \mathbb{E}_\tau\big[(\sum_t \gamma^t R_t^2)\big]/(1-\gamma)$. Rearranging the terms the thesis is proven. ■

It is important to notice that the factor $(1-\gamma)^2$ comes from the fact that the return variance is not normalized, unlike the reward volatility (intuitively, volatility measures risk on a shorter time scale). What is lost in the reward volatility compared to the return variance are the inter-temporal correlations between the rewards. However, Lemma 1 shows that the minimization of the reward volatility yields a low return variance. The opposite is clearly not true: as counterexample it is possible to consider a stock price, having the same value at the beginning and at the end of the investment period, but making complex movements in-between.

4 Risk-Averse Policy Gradient

In this section, we derive a policy gradient theorem for the reward volatility ν_π^2 and propose an unbiased gradient estimator. This will allow us to solve the optimization problem $\max_{\boldsymbol{\theta}\in\Theta} \eta_{\pi_{\boldsymbol{\theta}}}$ via stochastic gradient ascent. We introduce a volatility equivalent of the action-value function Q_π (Equation (1)) called *action-volatility* function, which is the volatility observed by starting from state s, taking action a, and following policy π thereafter:

$$X_\pi(s,a) := \mathop{\mathbb{E}}_{\substack{s_{t+1}\sim P(\cdot|s_t,a_t)\\ a_{t+1}\sim\pi(\cdot|s_{t+1})}}\left[\sum_{t=0}^{\infty}\gamma^t(\mathcal{R}(s_t,a_t)-J_\pi)^2|s,a\right], \tag{7}$$

Like the Q function, this can be written recursively by means of a Bellman equation:

$$X_\pi(s,a) = \big(R(s,a)-J_\pi\big)^2 + \gamma \mathop{\mathbb{E}}_{\substack{s'\sim P(\cdot|s,a)\\ a'\sim\pi_{\boldsymbol{\theta}}(\cdot|s')}}\big[X_\pi(s',a')\big]. \tag{8}$$

We define also the *state-volatility* function W_π as the expected value of X_π under the policy $\pi_{\boldsymbol{\theta}}$, i.e. the equivalent of the V function (Equation 2) for volatility. The linearity of this Bellman equation allows an alternative interpretation of the mean-volatility objective. In fact, by applying a reward transformation $R_\pi^\lambda(s_t,a_t) = R(s_t,a_t) - \lambda(R(s_t,a_t)-J_\pi)^2$, it is possible to formulate the problem as a standard RL problem, where X and W functions are reduced to Q and V. Nonetheless, R_π^λ is a non-stationary policy-dependent reward, hence it is not compliant with the usual MDP framework and it is not possible to apply standard value-based algorithms to it. In general, even policy gradient approaches cannot be used with this kind of rewards. However, with the obtained Bellman equation we can derive a Policy Gradient Theorem (PGT) that holds for both ν_π^2 and the transformed reward case, as done in [Sutton *et al.*, 2000] for the expected return:

Theorem 2 (Reward Volatility PGT) *Using the definitions of action-volatility and state-volatility function, the variance term ν_π^2 can be rewritten as:*

$$\nu_\pi^2 = (1-\gamma)\int_{\mathcal{S}} \mu(s) W_\pi(s) ds. \tag{9}$$

Moreover, for a given policy $\pi_{\boldsymbol{\theta}}$, $\boldsymbol{\theta}\in\Theta$:

$$\nabla\nu_\pi^2 = \mathop{\mathbb{E}}_{\substack{s\sim d_{\mu,\pi}\\ a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}}\Big[\nabla\log\pi_{\boldsymbol{\theta}}(a|s)X_\pi(s,a)\Big].$$

Sketch of Proof. Thanks to Equation (8), the computation of the gradients of X_π and W_π w.r.t. $\boldsymbol{\theta}$ is easily obtained:

$$\nabla X_\pi(s,a) = -2\big(R(s,a)-J_\pi\big)\nabla J_\pi + \gamma \mathop{\mathbb{E}}_{s'\sim P}\big[\nabla W_\pi(s')\big],$$

$$\begin{aligned}\nabla W_\pi(s) &= \nabla\int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s)X_\pi(s,a)\,\mathrm{d}a\\ &= \int_{\mathcal{A}}\left[\nabla\pi_{\boldsymbol{\theta}}(a|s)X_\pi(s,a) - 2\pi_{\boldsymbol{\theta}}(a|s)(R(s,a)-J_\pi)\nabla J_\pi\right]\mathrm{d}a\\ &\quad + \gamma\int_{\mathcal{S}}\left(\int_{\mathcal{A}} P(s'|s,a)\pi_{\boldsymbol{\theta}}(a|s)\,\mathrm{d}a\right)\nabla W_\pi(s')\,\mathrm{d}s'.\end{aligned}$$

By unrolling the recursive definition, we obtain the first right hand term in expectation under the $d_{\mu,\pi}$ and π. Its second component vanishes, allowing us to obtain the thesis from $\nabla\nu_\pi^2 = (1-\gamma)\int_{\mathcal{S}}\mu(s)\nabla W_\pi(s)\,\mathrm{d}s$. ■

The term that becomes null in the proof corresponds to the policy-dependent component of the reward. Therefore, we also proved that, in this special case, the PGT still applies after the transformation. With a simple extension it is possible to obtain the policy gradient theorem for the mean-volatility objective defined in equation (6). The action value and state value functions are obtained by combining the action value functions of the expected return (1) and of the volatility (7):

$$\begin{aligned}Q_\pi^\lambda(s,a) &:= Q_\pi(s,a) - \lambda X_\pi(s,a)\\ V_\pi^\lambda(s) &= V_\pi(s) - \lambda W_\pi(s).\end{aligned}$$

The policy gradient theorem thus states:

$$\nabla\eta_\pi = \mathop{\mathbb{E}}_{\substack{s\sim d_{\mu,\pi}\\ a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}}\Big[\nabla\log\pi_{\boldsymbol{\theta}}(a|s)Q_\pi^\lambda(s,a)\Big]. \tag{10}$$

4.1 Estimating the Risk-Averse Policy Gradient

To design a practical actor-only policy gradient algorithm, the action-value function Q_π needs to be estimated as in [Sutton and Barto, 1998; Peters and Schaal, 2008]. Similarly, we need an estimator for X_π. In this approximate framework, we consider to collect N finite trajectories $s_0^i, a_0^i, ..., s_{T-1}^i, a_{T-1}^i$, $i = 0, \dots, N-1$ per each policy update. An unbiased estimator of J_π can be defined as:

$$\hat{J} = \frac{1-\gamma}{1-\gamma^T}\frac{1}{N}\sum_{i=0}^{N-1}\sum_{t=0}^{T-1}\gamma^t \mathcal{R}_t^i, \tag{11}$$

where rewards are denoted as $\mathcal{R}_t^i = \mathcal{R}(s_t^i, a_t^i)$. This can be used to compute an estimator for the action-volatility function:

Lemma 3 *Let $\widehat{X}$ be the following estimator for the action-volatility function:*

$$\widehat{X} = \frac{1-\gamma}{1-\gamma^T}\frac{1}{N}\sum_{i=0}^{N-1}\sum_{t=0}^{T-1}\gamma^t \left[(\mathcal{R}_t^i - \hat{J}_1)(\mathcal{R}_t^i - \hat{J}_2)\right], \tag{12}$$

where $\hat{J}_1$ and $\hat{J}_2$, defined as in Equation (11)*, are taken from two different sets of trajectories $\mathcal{D}_1$ and $\mathcal{D}_2$, and a third set of samples $\mathcal{D}_3$ is used for the rewards $\mathcal{R}_t^i$ in Equation* (12)*. Then, $\widehat{X}$ is unbiased.*

Note that, in order to obtain an unbiased estimator for X, a *triple sampling* procedure is needed. This may be very restrictive. However, by adopting single sampling instead, the bias introduced is equivalent to the variance of $\widehat{J}$, so the estimator is still consistent. This result can be used to build a consistent estimator for the policy gradient $\nabla\eta_\pi$, as an extension of the PGT estimator [Sutton *et al.*, 2000]:

$$\begin{aligned}\widehat{\nabla}_N\eta_\pi =& \frac{1}{N}\sum_{i=0}^{N-1}\sum_{t=0}^{T-1}\gamma^t\Bigg(\sum_{t'=t}^{T-1}\gamma^{t'-t}\big[R_{t'}^i - \lambda\frac{1-\gamma}{1-\gamma^T} \\ &(R_{t'}^i - \widehat{J})^2\big]\Bigg)\nabla\log\pi_{\boldsymbol{\theta}}(a_t^i|s_t^i).\end{aligned} \tag{13}$$

5 Trust Region Volatility Optimization

In this section, we go beyond the standard policy gradient theorem and show it is possible to guarantee a monotonic improvement of the mean-volatility performance measure (6) at each policy update. Safe (in the sense of non-pejorative) updates are of fundamental importance when learning online on a real system; but also helps speeding up offline training by dynamically choosing the optimal step size. While the mean-volatility objective ensures a risk-averse *behavior* of the policy, the safe update ensures a risk-averse *update* of the parameters of the policy. Thus, if we care about the agent's performance within the learning process, we must consider the importance of the step sizes at each update of the parameters. Adapting the approach in [Schulman *et al.*, 2015] to our mean-volatility objective, we show it is possible to obtain a learning rate that guarantees that the performance of the updated policy is bounded with respect to the previous policy. The safe update is based on the advantage function, defined as the difference between the action value and state value function. From the linearity of the new Bellman equations, we can extend the definitions of advantage $A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s)$ to their λ-versions, to obtain the mean-volatility advantage function $A_\pi^\lambda(s,a)$. Furthermore, with the mean-volatility objective all the theoretical results leading to the *TRPO* algorithm hold. In particular, Theorem 4 is a λ-extension of Lemma 6.1 in [Kakade and Langford, 2002], with an interesting extra additive term[4]:

Theorem 4 (Performance Difference) *The performance difference between two policies π and $\widetilde{\pi}$ is equal to the sum of the expected mean-volatility advantage and a bonus term, related to the squared expected advantage:*

$$\begin{aligned}\eta_{\widetilde{\pi}} - \eta_\pi = \int_{\mathcal{S}} d_{\mu,\widetilde{\pi}}(s)\int_{\mathcal{A}}\widetilde{\pi}(a|s)A_\pi^\lambda(s,a)\,\mathrm{d}a\,\mathrm{d}s \\ +\lambda(1-\gamma)^2(J_{\widetilde{\pi}} - J_\pi)^2.\end{aligned} \tag{14}$$

Neglecting the last term, the bound becomes the same that could be obtained considering the transformed reward R_π^λ. In practice, it corresponds to considering the volatility of the previous policy rather than approximating the next one. This is, in general, the main issue that arises with the reward transformation: it works well for the on-policy case, but it cannot be handled with the same ease in the off-policy one. The aforementioned term adds a gain related to the square of the difference in the expected returns of the policies; therefore there is always a bonus w.r.t the reward transformation approach if the expected return of the second policy is either higher or lower than the first one. Following the approach proposed in [Schulman *et al.*, 2015], it is then possible to adopt an approximation $L_\pi^\lambda(\widetilde{\pi})$ of the surrogate function, which provides monotonic improvement guarantees by considering the KL divergence between the policies:

Theorem 5 (Safe Improvement Bound) *Consider the following approximation of $\eta_{\widetilde{\pi}}$, replacing the state-occupancy density of the old policy $d_{\mu,\pi}$:*

$$L_\pi^\lambda(\widetilde{\pi}) := \eta_\pi + \int_{\mathcal{S}} d_{\mu,\pi}(s)\int_{\mathcal{A}}\widetilde{\pi}(a|s)A_\pi^\lambda(s,a)\,\mathrm{d}a\,\mathrm{d}s; \tag{15}$$

Let

$$\alpha = D_{KL}^{max}(\pi,\widetilde{\pi}) = \max_s D_{KL}(\pi(\cdot|s),\widetilde{\pi}(\cdot|s))$$

$$\epsilon_\lambda = \max_s |\mathop{\mathbb{E}}_{a\sim\widetilde{\pi}}\left[A_\pi^\lambda(s,a)\right]|, \quad \epsilon = \max_s |\mathop{\mathbb{E}}_{a\sim\widetilde{\pi}}\left[A_\pi(s,a)\right]|$$

Then, the performance of $\widetilde{\pi}$ can be bounded as follows:[5]

$$\eta_{\widetilde{\pi}} \geq L_\pi^\lambda(\widetilde{\pi}) - \frac{2\gamma\epsilon_\lambda}{1-\gamma}\alpha + \lambda(1-\gamma)^2 M^2, \tag{16}$$

where

$$M := \max(0, A_\pi^{\widetilde{\pi}} - \frac{2\epsilon\gamma}{1-\gamma}\alpha, -A_\pi^{\widetilde{\pi}} - \frac{\gamma}{1-\gamma}\alpha R_{\max}),$$

$$A_\pi^{\widetilde{\pi}} := \int_{\mathcal{S}} d_{\mu,\pi}(s)\int_{\mathcal{A}}\widetilde{\pi}(a|s)A_\pi(s,a)\,\mathrm{d}a\,\mathrm{d}s.$$

[4] Different definitions result in different normalization terms.

[5] Comparing this bound to the results shown in the original paper, the denominator term is not squared due to return normalization.

Algorithm 1 Trust Region Volatility Optimization (TRVO)

```
Input: initial policy parameter $\boldsymbol{\theta}_0$, batch size $N$, number of iterations $K$, discount factor $\gamma$.
for $k = 0, \dots, K-1$ do
    Collect $N$ trajectories with $\boldsymbol{\theta}_k$ to obtain dataset $\mathcal{D}_N$
    Compute estimates $\widehat{J}$ as in Equation (11)
    Estimate advantage values $A^{\lambda}_{\boldsymbol{\theta}_k}(s, a)$
    Solve the constrained optimization problem
        $\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}\in\Theta} \left[L^{\lambda}_k(\boldsymbol{\theta}) - \frac{2\epsilon\gamma}{1-\gamma} D^{max}_{KL}(\pi_{\boldsymbol{\theta}_k}, \pi_{\boldsymbol{\theta}})\right]$
        where $\epsilon = \max_s \max_a |A^{\lambda}_{\boldsymbol{\theta}_k}(s, a)|$
        $L^{\lambda}_k(\boldsymbol{\theta}) = \eta_{\boldsymbol{\theta}_k} + \mathop{\mathbb{E}}_{\substack{s\sim d_{\mu,\pi_k} \\ a\sim\pi_{\boldsymbol{\theta}}(\cdot|s)}} A^{\lambda}_{\boldsymbol{\theta}_k}(s, a)$
end for
```

Finally, we can devise the first risk-averse trust-region optimization algorithm (to the best of our knowledge), which is called TRVO (Trust Region Volatility Optimization) and is outlined in Algorithm 1. The reader should notice that this extension is highly dependent on the the risk-measure chosen, and could not be easily applied to the other ones, which lack a linear Bellman equation [Sobel, 1982].

6 Related Works

Two streams of RL literature have been merged together in this paper: risk-averse objective functions and safe policy updates. As stated in [Di Castro *et al.*, 2012; García and Fernández, 2015], these two themes are related as they both reduce risk. The first is the reduction of inherent risk, generated by the stochastic nature of the environment, while the second is the reduction of model risk, related to the imperfect knowledge of model parameters. Several ways of minimizing inherent risk have been considered in RL literature. In particular, the *mean-variance* objective $J_\pi - \beta\sigma^2_\pi$ has been relevant especially in the financial field [Steinbach, 2001]. One of the earliest contributions for this specific objective is the risk-averse Bellman equation for the return variance defined in [Sobel, 1982]. This equation does not satisfy the monotonicity property of dynamic programming, preventing the use of value-based approaches. However, it is still possible, with actor only [Di Castro *et al.*, 2012] and actor-critic [Prashanth and Ghavamzadeh, 2014a] algorithms, to locally optimize this measure. In [Moody and Saffell, 2001] the authors maximize a reward-related risk measure, the *Sharpe ratio*, defined as the ratio between the mean and standard deviation of the reward. However their algorithm assumes no dependence between states and actions, which is true in simple trading environments (as the ones we consider in the experiments), but not in more realistic ones. Coherent risk measures represent another interesting alternative with favorable mathematical properties, for which it is possible to derive an actor-only policy gradient algorithm [Tamar *et al.*, 2015]. CVaR is the most frequently employed in finance and has been separately tackled with a distributional approach [Morimura *et al.*, 2010], and an actor-critic algorithm [Chow *et al.*, 2017]. Even if for some of these measures a Bellman Equation can be derived, unfortunately, such recursive relationships are always non-linear. This prevents one to extend safe guarantees (Theorem 5) and the properties of the TRPO algorithm to the mentioned risk measures. Finally, it is possible to introduce risk averse objectives in value function based algorithms such as Q-learning [Tamar *et al.*, 2016], but it is limited to discrete action spaces.

The second literature stream is dedicated to the safe update, which, until now, has only been defined for the standard risk-neutral objective function. The seminal paper for this setting is [Kakade and Langford, 2002], which proposes a conservative policy iteration algorithm with monotonic improvement guarantees for mixtures of greedy policies. This approach is generalized to stationary and stochastic policies in [Pirotta *et al.*, 2013b; Schulman *et al.*, 2015]. Building on the former, monotonically improving policy gradient algorithms are devised for Gaussian, Lipschitz, and more recently, smoothing policies [Pirotta *et al.*, 2013a; Papini *et al.*, 2017; Pirotta *et al.*, 2015; Papini *et al.*, 2019]. On the other hand, [Schulman *et al.*, 2015] propose TRPO, a general policy gradient algorithm inspired by the monotonically-improving policy iteration strategy, which enjoyed great empirical success in recent years, especially in combination with deep policies.

7 Experiments

In this section, we show an empirical analysis of the performance of TRVO (Algorithm 1) applied in two financial trading tasks: the first on an equity index, the S&P 500, and the second on spot Foreign Exchange (FX): USD/EUR and USD/JPY. The first baseline we compare to is a mean-variance policy gradient approach presented in [Di Castro *et al.*, 2012] (indicated as MV-PG), which we adjusted to take into account discounting.The second one is Direct Reinforcement Learning (DRL) [Moody and Saffell, 2001]. Finally we consider a risk averse transformation of the rewards, $\widetilde{R}_t := (1 - \exp\{-cR_t\})/c$, in the original *TRPO* algorithm (indicated as TRPO-exp). It represents a first-order approximation of mean-volatility, but it is sound only for small values of the risk-aversion coefficient, since negative rewards can generate strong instabilities of the learning process. As shown below, TRVO is capable of obtaining a complete Pareto frontier on both these environments and it converges sooner than the baselines.

7.1 S&P 500 Trading

This first environment considers the daily prices of the S&P index from the 1980s, until 2019. The possible actions are $a_t \in \{-1, 0, 1\}$, where -1 indicates a short, 1 a long, and 0 a flat position (thus, short selling is possible). We assume that at each time-step we go long or short of the same unitary amount, thus the profits (and losses) are not re-invested, which means that the final gain is the sum of all the rewards. The value of the asset at time t is p_t, and the reward is equal to $R_t = a_t(p_t - p_{t-1}) - f|a_t - a_{t-1}|$, where the first term is the profit or loss given by the action a_t, and the second term represents the transaction costs, where f is the proportionality constant, set to $7 \cdot 10^{-5}$. The policy we used is a neural

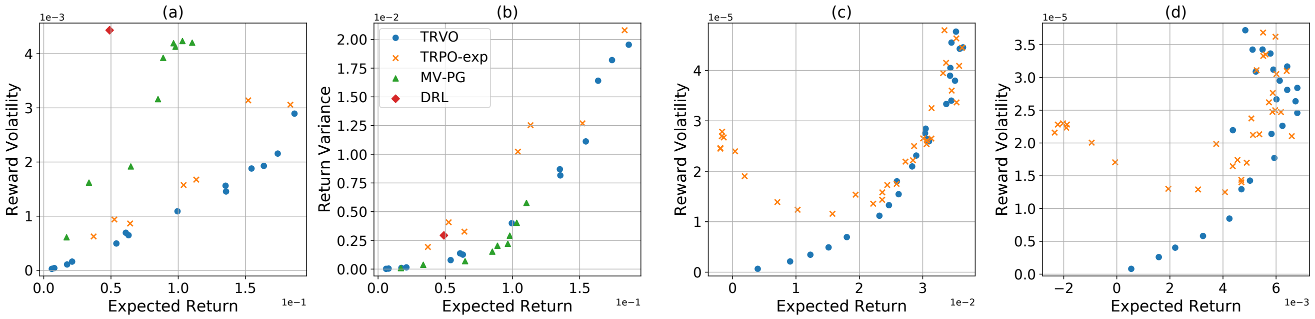

Figure 1: (a) and (b): expected return, reward volatility, return variance in the S&P 500 environment with: TRVO, TRPO-exp, MV-PG, DRL; (c) and (d): expected return, mean volatility in the FX environment in training (c) testing (d). Performance is on 3 months and not normalized.

network with two hidden layers and 64 neurons per hidden layer. The state consists of the last 10 days of percentage price changes, the previous portfolio position and the fraction of episode left (50 days long).

Results. The relevant plots for this environment are the first two in Figure 1, obtained on in-sample data. Plot (a) shows the Pareto frontier obtained with the four different algorithms by changing the risk aversion coefficient in the mean-volatility space, plot (b) in the mean-variance space. It is evident that the frontier generated by TRVO dominates the naive approach (TRPO-exp). Also, TRPO-exp becomes unstable for high levels of the risk-aversion parameter c, so it is not possible to find the value for which the risk aversion is maximal, which is why there are no points in the bottom left. The same figure includes also the results obtained with MV-PG, trained with the same number of iterations as TRVO (and TRPO-exp).In the mean-volatility space (Figure 1.a), the frontier generated by TRVO is clearly dominating. Instead, in the mean-variance space (Figure 1.b), the frontiers generated by MV-PG and TRVO are overlapping, but while the points generated by TRVO span a wide part of the space, those generated by MV-PG are concentrated in the lower-leftmost part of the graph even though they are trained with different risk aversions. This is due in part to the fact that MV-PG has not reached convergence even though it was given the same number of steps as TRVO, and reflects the faster convergence of TRPO w.r.t. GPOMDP. For DRL it is not possible to set the risk-aversion, hence it consists in a single point, which is on the Mean-Variance frontier, but it is instead dominated w.r.t. the Mean-Volatility criterion.

7.2 FX Trading

In the second experiment, actions and rewards are defined in the same way as before, but two different assets are considered: the FX rates USD/EUR and USD/JPY. The dataset has a much higher frequency (one datapoint per minute), hence also the agent can act every minute for a total of 1170 steps per episode (a trading day). The possible actions correspond to the position to keep for each asset, and the fee for each transaction is $f = 10^{-6}$. The training has been performed for a total of $5 \cdot 10^7$ steps on the 2017 dataset, while the testing was applied on 2018.

Results. The results for this environment can be found in the last two plots in Figure 1. We can see that TRPO-exp obtains the same results as TRVO for small risk-aversion coefficients, both in training (c) and in testing (d). However, higher coefficients lead to instability in the exponential reward, that is gradually dominated by TRVO. It is interesting to notice that the settings having small or null risk-aversion coefficients (top right of the plots) are on the edge of the frontier in training, but are dominated in testing by more risk-averse policies. In this environment, MV-PG converges to a sub-optimal policy with null expected return, while DRL does not improve. Hence, they are not shown in the figures.

8 Conclusions

We proposed a novel methodology for risk-averse RL, exploiting, for the first time, a safe improvement bound. This was possible thanks to the definition of a risk measure called reward volatility that captures the variability of the rewards between steps. Optimizing this measure allows to obtain smoother trajectories that avoid shocks, which is a fundamental feature in a trading setting, and has never been considered by other risk measures so far. We showed interesting theoretical properties of reward-volatility: it bounds the variance of the returns and, differently from other risk measures, it has a linear Bellman equation. A policy gradient theorem for the mean-volatility objective was derived and, thanks to the aforementioned linearity, we obtained TRVO, a trust region algorithm that exploits a monotonic improvement bound of our objective. The proposed algorithm was tested on two financial trading environments where it was shown to outperform the baselines, obtaining better Pareto frontiers in shorter time. This work lays the foundation for extensions to both off-policy and online settings. To conclude, the developed framework is the first to take into account two kinds of safety, as it is capable of keeping risk under control while maintaining the same training and convergence properties as state-of-the-art risk-neutral approaches.

Acknowledgements

The research was conducted under a cooperative agreement between ISI Foundation, Banca IMI and Intesa Sanpaolo Innovation Center.

References

[Chow *et al.*, 2017] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *JMLR*, 18(1):6070–6120, 2017.

[Deisenroth *et al.*, 2013] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

[Di Castro *et al.*, 2012] Dotan Di Castro, Aviv Tamar, and Shie Mannor. Policy gradients with variance related risk criteria. *ICML*, 1, 06 2012.

[García and Fernández, 2015] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *JMLR*, 16:1437–1480, 2015.

[Heess *et al.*, 2017] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, S. M. Ali Eslami, Martin A. Riedmiller, and David Silver. Emergence of locomotion behaviours in rich environments. *CoRR*, abs/1707.02286, 2017.

[Kakade and Langford, 2002] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274, 2002.

[Moody and Saffell, 2001] John Moody and Matthew Saffell. Learning to trade via direct reinforcement. *IEEE transactions on neural Networks*, 12(4):875–889, 2001.

[Morimura *et al.*, 2010] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *ICML*, 2010.

[OpenAI, 2018 accessed May 2020] OpenAI. Openai five. https://blog.openai.com/openai-five/, 2018 - accessed May 2020.

[Papini *et al.*, 2017] Matteo Papini, Matteo Pirotta, and Marcello Restelli. Adaptive batch size for safe policy gradients. In *NeurIPS*, pages 3591–3600, 2017.

[Papini *et al.*, 2019] Matteo Papini, Matteo Pirotta, and Marcello Restelli. Smoothing policies and safe policy gradients, 2019.

[Peters and Schaal, 2008] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

[Pirotta *et al.*, 2013a] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Adaptive step-size for policy gradient methods. In *NeurIPS 26*, pages 1394–1402. Curran Associates, Inc., 2013.

[Pirotta *et al.*, 2013b] Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *ICML*, pages 307–315, 2013.

[Pirotta *et al.*, 2015] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2-3):255–283, 2015.

[Prashanth and Ghavamzadeh, 2014a] L. A. Prashanth and Mohammad Ghavamzadeh. Actor-critic algorithms for risk-sensitive reinforcement learning. *arXiv preprint arXiv:1403.6530*, 2014.

[Prashanth and Ghavamzadeh, 2014b] L. A. Prashanth and Mohammad Ghavamzadeh. Variance-constrained actor-critic algorithms for discounted and average reward mdps. *CoRR*, abs/1403.6530, 2014.

[Schulman *et al.*, 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, pages 1889–1897, 2015.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[Shen *et al.*, 2014] Yun Shen, Ruihong Huang, Chang Yan, and Klaus Obermayer. Risk-averse reinforcement learning for algorithmic trading. pages 391–398, March 2014.

[Sobel, 1982] Matthew J. Sobel. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19(4):794–802, 1982.

[Steinbach, 2001] Marc C Steinbach. Markowitz revisited: Mean-variance models in financial portfolio analysis. *SIAM review*, 43(1):31–85, 2001.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[Sutton *et al.*, 2000] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NeurIPS*, pages 1057–1063, 2000.

[Tamar and Mannor, 2013] Aviv Tamar and Shie Mannor. Variance adjusted actor critic algorithms. *arXiv preprint arXiv:1310.3697*, 2013.

[Tamar *et al.*, 2015] Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy Gradient for Coherent Risk Measures. *CoRR*, page 9, 2015.

[Tamar *et al.*, 2016] Aviv Tamar, Dotan Di Castro, and Shie Mannor. Learning the variance of the reward-to-go. *The Journal of Machine Learning Research*, 17(1):361–396, 2016.

[Tamar *et al.*, 2017] Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Sequential Decision Making With Coherent Risk. *IEEE Transactions on Automatic Control*, 62(7):3323–3338, July 2017.