Risk-Sensitive Reinforcement Learning: a Martingale Approach to Reward Uncertainty

Nelson Vadori, Sumitra Ganesh,
Prashant Reddy, Manuela Veloso

Portfolio optimization example environment

Brain, Mind and Markets Laboratory

	LowVol	MediumVol	HighVol
Action pair: (q_t^{RF}, q_t^{R})	$\mu = 0.2, \sigma = 0.5$	$\mu = 0.6, \sigma = 1$	$\mu = 1, \sigma = 1.5$

- $R_{t+1} = R_{t+1}^{RF} + R_{t+1}^{R}$
  - Risk-free reward: $R_{t+1}^{RF} = q_t^{RF}\mu(s_t)$
  - Risky reward: $R_{t+1}^{R} = q_t^{R}(\mu(s_t) + \sigma(s_t)h_{t+1})$
- $q_t^{RF}, q_t^{R} \geq 0, q_t^{RF}, q_t^{R} \in Z$
- Budget constraint: $q_t^{RF} + q_t^{R} \leq q_{max} = 5$
- $\Rightarrow 21\ possible\ actions$

- State transition matrix is designed such that the more we invest in the risky asset, the more likely we are to reach a higher volatility state.

Table 2: Section 5.2 Portfolio Optimization: state transition matrix as a function of the chosen action (quantity q_t^R invested in the risky asset)

$q_t^R = 5$	LowVol	MediumVol	HighVol
LowVol	0.05	0.25	0.7
MediumVol	0.05	0.25	0.7
HighVol	0.05	0.25	0.7

$2 < q_t^R < 5$	LowVol	MediumVol	HighVol
LowVol	0.1	0.45	0.45
MediumVol	0.1	0.45	0.45
HighVol	0.1	0.45	0.45

$0 < q_t^R \leq 2$	LowVol	MediumVol	HighVol
LowVol	1/3	1/3	1/3
MediumVol	1/3	1/3	1/3
HighVol	1/3	1/3	1/3

$q_t^R = 0$	LowVol	MediumVol	HighVol
LowVol	0.5	0.45	0.05
MediumVol	0.5	0.45	0.05
HighVol	0.5	0.45	0.05

Brain, Mind and Markets Laboratory

Simulation

- I ran simulations with four different seeds. In each case, the two algorithms (CMV and sample avg).
- No of episodes = 10000, no of steps per ep = 20, exploration in the first 1000 exponential decay, alpha_learning_rate=0.1.
- Discount rate gamma: CMV=1.0, sample_avg=0.9
- The results are consistent across different seeds.

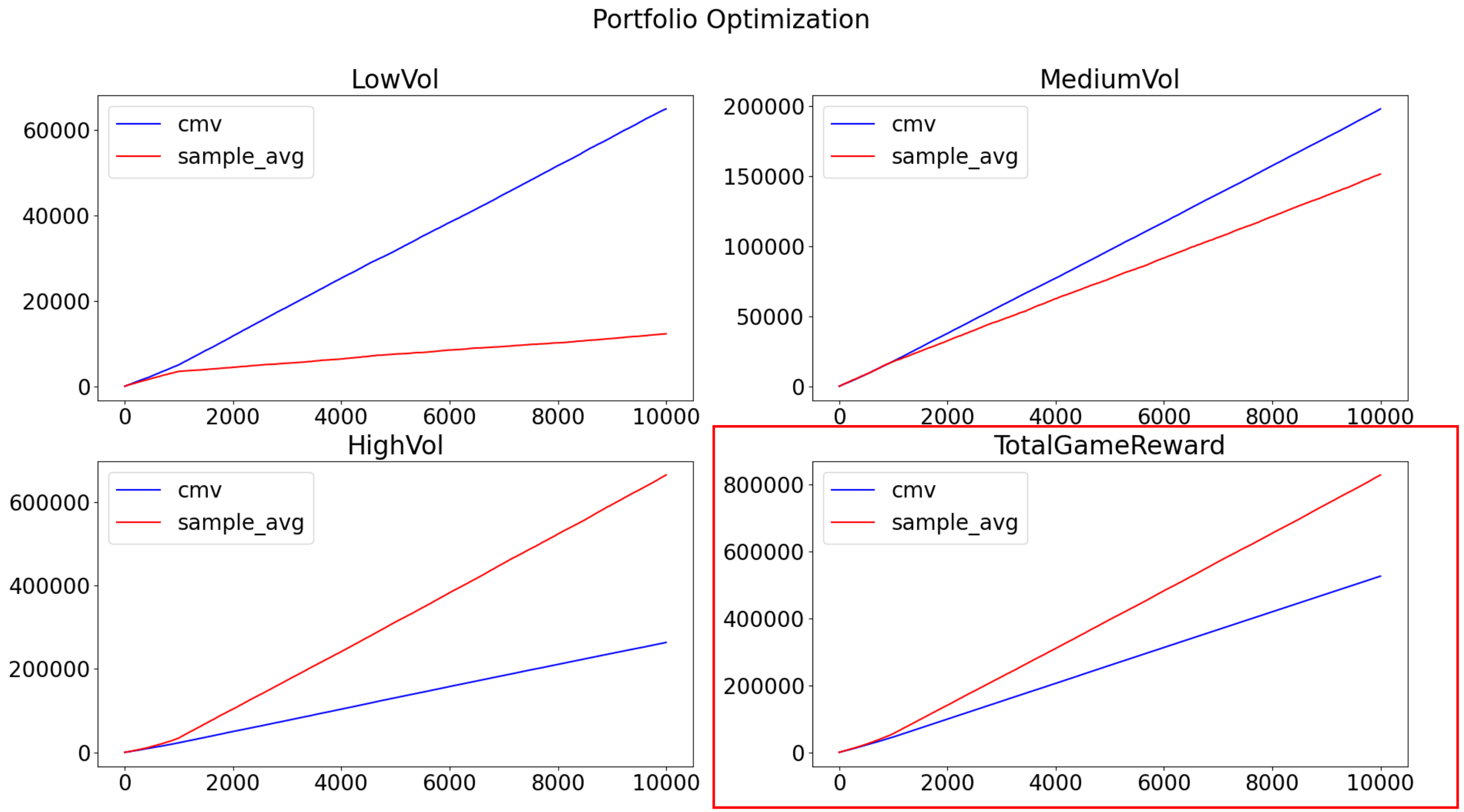
(q_rf, q_r) = risk_free quantity, risky quantity)

- There is no unallocated capital in the sample avg method.
- The sample avg agent seems to be very "risk seeking". However, given the environment setting, I think this is the optimal thing to do.
- The Chaotic Mean Variance method seems to be "risk-averse", and do not allocate 100% capital.

seed=5	LowVol	MediumVol	HighVol
CMV	(3, 2)	(4, 1)	(4 , 0)
sample_avg	(0, 5)	(0, 5)	(0, 5)
seed=234	LowVol	MediumVol	HighVol
CMV	(4, 1)	(3, 1)	(4, 0)
sample_avg	(0, 5)	(0, 5)	(0, 5)
seed = 154	LowVol	MediumVol	HighVol
CMV	(5, 0)	(4, 0)	(4 , 0)
sample_avg	(0, 5)	(0, 5)	(0, 5)
seed = 567	LowVol	MediumVol	HighVol
CMV	(3, 2)	(4, 1)	(3, 1)
sample_avg	(0, 5)	(0, 5)	(0, 5)

Simulation results (seed 567)

Brain, Mind and Markets Laboratory



- The sample avg agent outperforms the CMV agent in general, based on the running sum of rewards received by the agent, regardless of which state they are at.
- This is mostly due to the fact that the sample avg agent allocate 100% in the risky asset.
- On avg, it is better to stay in the HighVol case (mu=1)

Motivation

- Risk-sensitive RL commonly uses variance as a measure of risk.
- The common goal is to take into account the distribution of the cumulative rewards in order to learn a variety of policies, usually parameterized by a risk parameter such as the mean-variance trade-off, the CVaR percentile or an upper bound on variance.
- However, this could be problematic.
- E.g. Often the learned policies typically lead to the distribution of cumulative rewards having lower mean but also lower variance.

Claim and contribution

Brain, Mind and Markets Laboratory

- In a stochastic environment where the reward process is also stochastic, one should separate the randomness in the reward:
  - “predictable part”: where the reward randomness is due to state transition
  - “chaotic part”: where the reward randomness is due to the randomness nature of the reward process itself. If the chaotic part is 0, then the reward process is deterministic.
- I summarized the above from the paper (introduction). This is exactly what we have in our paper.
- They used Doob decomposition method to separate the two.
- The proofs look complicated (as usual) but the algorithm looks okay.

Motivation example environment

Brain, Mind and Markets Laboratory

	S0	S1
A0	2	10
A1	$4 + \sigma h_t$	$8 + \sigma h_t$

- $P(S_{t+1}|S_t) = 50\%$
- $\sigma \geq 0 \ (meta = 0.16)$
- $h_t \sim N(0,1)$
- A0: risk-free investment
- A1: risky investment
- Optimal policy: (S0, A1) (S1, A0)

Portfolio optimization example environment

Brain, Mind and Markets Laboratory

	LowVol	MediumVol	HighVol
Action pair: (q_t^{RF}, q_t^{R})	$\mu = 0.2, \sigma = 0.5$	$\mu = 0.6, \sigma = 1$	$\mu = 1, \sigma = 1.5$

- $R_{t+1} = R_{t+1}^{RF} + R_{t+1}^{R}$
  - Risk-free reward: $R_{t+1}^{RF} = q_t^{RF}\mu(s_t)$
  - Risky reward: $R_{t+1}^{R} = q_t^{R}(\mu(s_t) + \sigma(s_t)h_{t+1})$
- $q_t^{RF}, q_t^{R} \geq 0, q_t^{RF}, q_t^{R} \in Z$
- Budget constraint: $q_t^{RF} + q_t^{R} \leq q_{max} = 5$
- $\Rightarrow 21\ possible\ actions$

- State transition matrix is designed such that the more we invest in the risky asset, the more likely we are to reach a higher volatility state.

Table 2: Section 5.2 Portfolio Optimization: state transition matrix as a function of the chosen action (quantity q_t^R invested in the risky asset)

$q_t^R = 5$	LowVol	MediumVol	HighVol
LowVol	0.05	0.25	0.7
MediumVol	0.05	0.25	0.7
HighVol	0.05	0.25	0.7

$2 < q_t^R < 5$	LowVol	MediumVol	HighVol
LowVol	0.1	0.45	0.45
MediumVol	0.1	0.45	0.45
HighVol	0.1	0.45	0.45

$0 < q_t^R \leq 2$	LowVol	MediumVol	HighVol
LowVol	1/3	1/3	1/3
MediumVol	1/3	1/3	1/3
HighVol	1/3	1/3	1/3

$q_t^R = 0$	LowVol	MediumVol	HighVol
LowVol	0.5	0.45	0.05
MediumVol	0.5	0.45	0.05
HighVol	0.5	0.45	0.05

Harvey notes

- Their environment is gaussian only. In their motivation case, we know that our sample average method will work.
- I have to dig into the portfolio optimization case, but with a gaussian process I think our method should work as well.

Brain, Mind and Markets Laboratory

Chaotic Mean-Variance Q-learning (episodic case)

Algorithm 1 **CMV-Q-Learning (episodic case)**

Input: Q^{β}-table initialized arbitrarily, learning rate $(\alpha_t)_{t\geq 0}$, $\overline{R}(s,a)$ and $N(s,a)$ initialized to 0.

Output: optimal policy $\pi_*(s) = \mathrm{argmax}_a Q^{\beta}_*(s,a)$

for each episode **do**

  initialize $s_t = s_0$

  while s_t is not terminal **do**

    Choose a_t from s_t using a policy derived from Q^{β} (e.g. ϵ-greedy).

    Take action a_t, observe s_{t+1}, R_{t+1}.

    $N(s_t,a_t) \leftarrow N(s_t,a_t)+1$; $\overline{R}(s_t,a_t) \leftarrow \overline{R}(s_t,a_t) + \frac{1}{N(s_t,a_t)}\underline{(R_{t+1}-\overline{R}(s_t,a_t))}$

    $Q^{\beta}(s_t,a_t) \leftarrow (1-\alpha_t)Q^{\beta}(s_t,a_t) + \alpha_t(\underline{R_{t+1} - \boxed{\tfrac{\beta}{2}}(R_{t+1} - \overline{R}(s_t,a_t))^2} + \max_a Q^{\beta}(s_{t+1},a))$

    $s_t \leftarrow s_{t+1}$

N: number of steps

Q-update

They set the discount rate gamma=1

They call this "chaotic" component. (The "surprise" part of the reward process)

Risk-adjusted reward ($\beta \geq 0$)

Harvey notes

Algorithm 1 CMV-Q-Learning (episodic case)

Input: Q^β-table initialized arbitrarily, learning rate $(\alpha_t)_{t\geq 0}$, $\overline{R}(s,a)$ and $N(s,a)$ initialized to 0.

Output: optimal policy $\pi_*(s) = \text{argmax}_a Q^\beta_*(s,a)$

for each episode **do**
  initialize $s_t = s_0$
  while s_t is not terminal **do**
    Choose a_t from s_t using a policy derived from Q^β (e.g. ϵ-greedy).
    Take action a_t, observe s_{t+1}, R_{t+1}.
    $N(s_t,a_t) \leftarrow N(s_t,a_t)+1$; $\overline{R}(s_t,a_t) \leftarrow \overline{R}(s_t,a_t) + \frac{1}{N(s_t,a_t)}(R_{t+1} - \overline{R}(s_t,a_t))$
    $Q^\beta(s_t,a_t) \leftarrow (1-\alpha_t)Q^\beta(s_t,a_t) + \alpha_t(R_{t+1} - \boxed{\frac{\beta}{2}}(R_{t+1} - \overline{R}(s_t,a_t))^2 + \max_a Q^\beta(s_{t+1},a))$
    $s_t \leftarrow s_{t+1}$

This is going to be large overtime

Overtime, the algorithm will find some mean. I don't think in the lepto case, this is going to converge to the true mean.

In a lepto case, this is not going to work, the square of the reward is going to explode the reward calculation.

Other parameters

- Learning rate: $\alpha = 0.1$
- Rollout episodes = $5 * 10^4$
- Timestep = 20
- $\beta = 0.0, 0.2, 0.5, 3.0$