A short discussion of stochastic rewards in RL environment

By Harvey Huang

Stochastic rewards in RL environments

Brain, Mind and Markets Laboratory

- Noises on rewards (Strens, 2000):
  - “This greedy policy fails when rewards are uncertain; the system can converge to a local minimum where the greedy behavior is sub-optimal because its decisions are based on maximum likelihood estimates (e.g. averages) which are not equal to the true parameters.”
  - Strens (2000) proposed a Bayesian approach.
  - The noises on rewards are assumed to be gaussian.
- Noises on rewards (Moreno et al., 2006)
  - Moving (sample) average method.
  - The noises on rewards are gaussian or impulsive.

Stochastic rewards in RL environments

- Corrupted rewards (Everitt et al., 2017)
  - Observed reward is drawn from a “corruption” function given the true reward and state
    - reward = C(s, true_reward)

- Corrupted and stochastic rewards (Romoff et al., 2018)
  - Gaussian noises on rewards
  - Uniform noises on rewards
  - Sparse rewards
  - They used sample average
    - “Sample avg does not improve performance much if rewards are delayed (e.g. board game)”

Stochastic rewards in RL environments

Brain, Mind and Markets Laboratory

- Perturbed rewards (Wang et al. 2020)
  - Reward DGP falls into the same category: gaussian .
  - They realized it is better to use unbiased estimator for true reward.
  - However, they used a complicated approach adopted from supervised learning.
- In a nutshell
  - We need to find an approximation of the true reward function.
    - Many existing approaches: approximate reward function -> find mean -> learn the optimal policy.
  - I think this is somehow influence by the model-based RL research in CS.
  - However, to find optimal policy we need only the mean reward.
    - Find the mean (a reward function is not necessary) -> learn the optimal policy

Robust RL to stochastic rewards

Brain, Mind and Markets Laboratory

- Many existing approaches to tackle robustness focus on reducing the second moment (variance).
  - In theory/proofs
  - In standard games/environment simulations as well.
- Analysis/proof of convergence requires strict and tight assumptions
  - E.g. uniform/gaussian rewards
- What about other environments where we have no idea what the reward generating process is (particularly if rewards are note gaussian)?
  - And what is the best way to get the “unbiased (and efficient) estimates”
- This is the issue we intend to raise (and to provide elegant solutions).

Risk-sensitive RL (Vadori et al., 2020)

Brain, Mind and Markets Laboratory

- Risk-sensitive RL commonly uses variance as a measure of risk.
- The common goal is to consider 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.

Risk-sensitive RL (Vadori et al., 2020)

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.
- Approach: doob decomposition method to separate the two.
- This is also an important message we want to deliver
  - We should separate the randomness due to different sources.
- Our approach: distributional RL combined with efficient statistics.

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

Combine everything together

Brain, Mind and Markets Laboratory

- In a stochastic environment with stochastic rewards, regardless of the reward DGP (specifically, not limited to gaussian/uniform distribution), how do agent perform well?
  - By finding the optimal policy (optimal action in a particular state).
- i.e. agents need to find out, ***on average***, which action is better in a state.
  - ➔ find ***unbiased*** mean reward ***efficiently***.
  - There are solutions from statistics to deal with all kinds of stochastic reward distributions. One doesn't need to build sophisticated and specially designed approaches.
- Approach that we use: you can use statistic tools to improve the robustness of RL agents.
  - We should find unbiased and efficient mean reward to handle reward uncertainty.