Uncertainty Decomposition for Risk Control in Automated Stock Trading

Anita Mahinpei

Harvard University

Introduction

One natural application of reinforcement learning is automated stock trading and portfolio management. A simple formulation of stock trading as an RL problem would aim to maximize the change in the portfolio value. This approach, however, will not be appropriate for all individuals. Depending on their financial circumstances, some individuals would prefer more conservative portfolios that do not maximize returns but have lower risk. Others might prefer high risk portfolios that have lower expected returns but have the potential (albeit with low probability) for much greater returns over time. Risk assessment is crucial for such portfolio customization. More generally, risk assessment can help improve the exploration exploitation trade-off by ensuring that exploratory actions are not too risky.One approach to risk aversion could be to use a measure of uncertainty in the evaluated value functions. A more risk tolerant portfolio should be more likely to take actions with high uncertainty/variance in their expected return while more conservative portfolios should be less likely to select such actions even if the expected return is high. This project investigates the suitability of risk control via uncertainty estimation for stock trading problems by building on the uncertainty decomposition algorithm developed by Clements et al [1].

Related Works

Various approaches have been proposed for estimating uncertainty in different reinforcement learning algorithms for both model based and model free problems. Leno et al. use bootstrapped neural networks to estimate the variance in the Q-function and request human intervention when the variance/uncertainty is above a threshold [3]. Tamar et al. also use the mean-variance trade-off for risk assessment but in the context of policy gradient algorithms [6]. O'Donoghue et al. propose the Uncertainty Bellman Equation for propagating local uncertainty estimates to global estimates and suggest various techniques for approximating local uncertainties [5].

These approaches do not attempt to disentangle aleatoric and epistemic uncertainties in their estimates. Aleatoric uncertainty is uncertainty that is inherent to the data and stems from stochasticity of the environment or our observations of the environment while epistemic uncertainty stems from limitations in our model or training data. Disentangling these sources of uncertainty allows the algorithm to avoid states with high aleatoric uncertainty for risk aversion purposes while not limiting exploration too much by ensuring that states with high epistemic uncertainty are not penalized as much. Clements et al. tackled the problem of disentangling aleatoric and epistemic uncertainties in Deep Q Learning. Rather than learning a single point estimate, they learned several quantiles of the Q-function distribution and used these quantiles to estimate aleatoric and epistemic uncertainties [1]. They showed that their algorithm was capable of avoiding the cliff in a grid world and that the probability of finding the safer route could be tuned with their risk aversion parameter [1].

Methodology

This project focuses on managing a single stock portfolio where we are limited to buying or selling a single share at each time step. The problem is formulated as a Markov Decision Process (MDP) with the following components:

1. **State:** the state contains the agent's remaining balance, the number of shares owned, the stock price, and a set of common market indicators from StockStats's pandas DataFrame wrapper [7].
2. **Action:** the agent is limited to either buying a single new share, selling one of their shares, or holding all their shares at each time step.
3. **Reward:** the immediate reward is the change in the total portfolio value (total stock value + remaining balance) at the end of a trading day.

Uncertainty Aware Deep Q Learning:
The Q-function is approximated by a deep neural network that takes the state as input and outputs N quantile estimates of the action-value distribution for each of the three plausible actions.

Methodology (cont.)

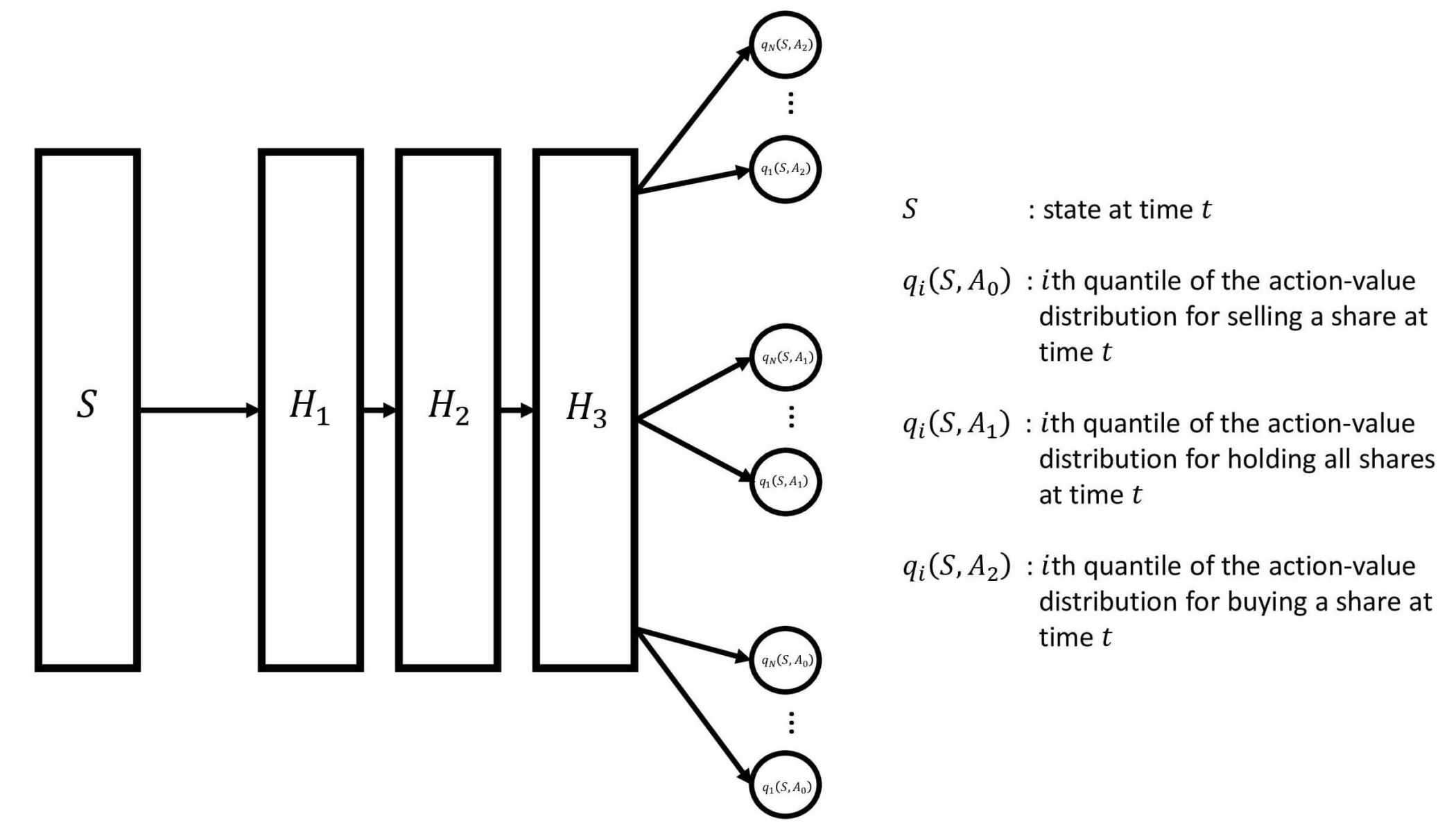

S
H_1
H_2
H_3
$q_N(S,A_2)$
$q_1(S,A_2)$
$q_N(S,A_1)$
$q_1(S,A_1)$
$q_N(S,A_0)$
$q_1(S,A_0)$
S : state at time t
$q_i(S,A_0)$: ith quantile of the action-value distribution for selling a share at time t
$q_i(S,A_1)$: ith quantile of the action-value distribution for holding all shares at time t
$q_i(S,A_2)$: ith quantile of the action-value distribution for buying a share at time t

Fig. 1: Uncertainty Aware Deep Q Learning Neural Network Architecture

The network is trained using the quantile regression loss function where $Z(s,a)$ is the distribution of returns parameterized by N quantiles $\tau_i = \frac{i}{N+1}$ with values q_i. For TD learning, the distribution is replaced by the Bellman target $R(s,a) + \gamma \max_{a'} Z(s',a')$ [2]:

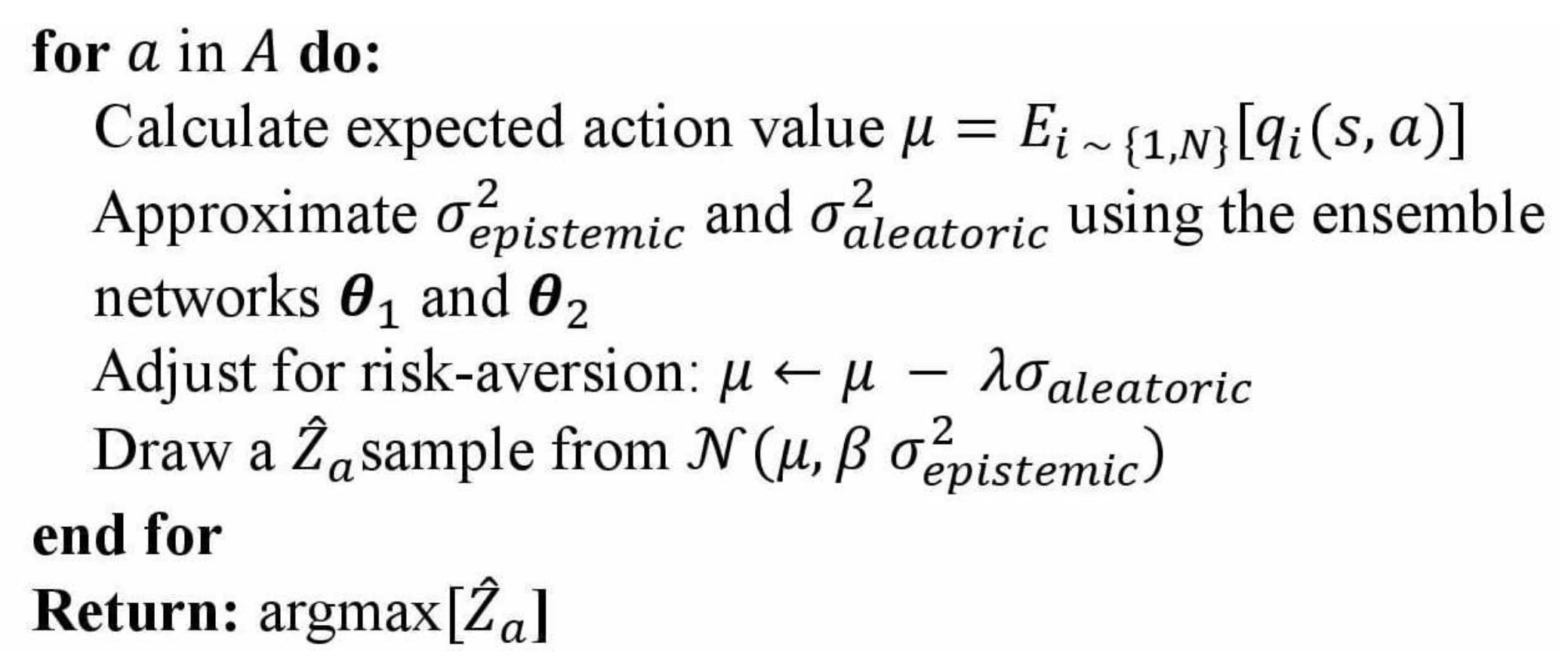
$$\mathcal{L}(\boldsymbol{q}) = \mathbb{E}_{z\sim Z(s,a)} \sum_{i=1}^{N} \rho_{\tau_i}(z - q_i(s,a))$$

$$\text{where } \rho_{\tau_i}(x) = x(\tau_i - \mathbb{1}_{x<0})$$

Clements et al. define epistemic uncertainty as the expected variance in the quantile estimations and aleatoric uncertainty as the variance in the expected quantile values. They show however, that these two uncertainties can be approximated using only an ensemble of two DQNs with parameters $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ [1]:

$$\tilde{\sigma}^2_{\text{epistemic}} = \frac{1}{2}\mathbb{E}_{i\sim\{1,N\}}[q_i(\boldsymbol{\theta_1},s,a) - q_i(\boldsymbol{\theta_2},s,a)]^2$$
$$\tilde{\sigma}^2_{\text{aleatoric}} = \text{cov}_{i\sim\{1,N\}}(q_i(\boldsymbol{\theta_1},s,a), q_i(\boldsymbol{\theta_2},s,a))$$

The action selection algorithm is then defined as follows [1]:



```
for $a$ in $A$ do:
    Calculate expected action value $\mu = E_{i\sim\{1,N\}}[q_i(s,a)]$
    Approximate $\sigma^2_{epistemic}$ and $\sigma^2_{aleatoric}$ using the ensemble
    networks $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$
    Adjust for risk-aversion: $\mu \leftarrow \mu - \lambda\sigma_{aleatoric}$
    Draw a $\hat{Z}_a$sample from $\mathcal{N}(\mu, \beta\,\sigma^2_{epistemic})$
end for
Return: argmax$[\hat{Z}_a]$
```

Financial Turbulence Index (FTI):
FTI is one measure of extreme asset fluctuation used by the stock trading library FinRL for risk control [4]. In this project, a DQN model that uses FTI for risk assessment is also built for comparison with the UA-DQN model. All buying is halted when FTI is larger than a pre-defined threshold and shares are gradually sold until FTI drops below the tolerance threshold.

Results and Discussion

The models were trained and evaluated with historical data from Yahoo Finance for eight relatively volatile stocks. The data from 2008 to 2015 were used for training while the data from 2016 to 2020 were used for testing. Two sample plots of the total returns as a fraction of initial assets for the test data are presented below. The performance of a baseline model that always buys shares is also plotted for comparison. In general, increasing the risk penalty λ seems to improve total returns. The UA-DQN model tends to give the best performance as evaluated by total returns on the test data. The experimental results suggest that uncertainty decomposition can be a suitable technique for risk control in automated stock trading. Further experiments are needed to make conclusive remarks.

Results and Discussion (cont.)

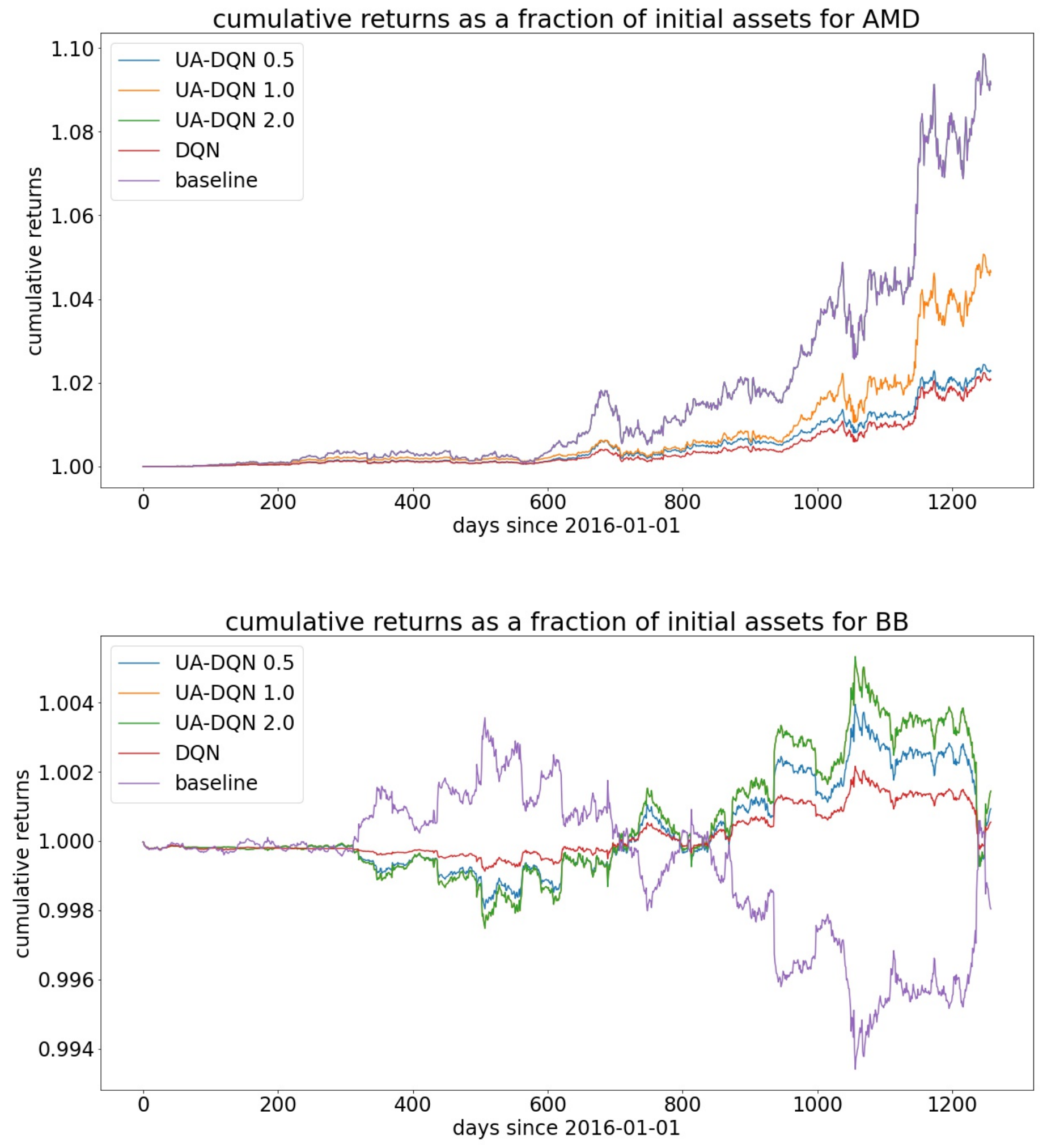


Fig. 3: Plots of cumulative returns as a fraction of initial assets over time for AMD and BlackBerry stocks for a baseline, DQN, and three UA-DQN models with different risk penalty parameters (λ).

In the future, experiments should be performed to see if uncertainty decomposition is necessary for good performance or if a simple measure of total uncertainty can also produce similar results. Deep Q-Learning is not the most suitable RL algorithm for stock trading [4]. Future experiments should try to decompose uncertainties of algorithms such as Proximal Policy Optimization which are more suitable for realistic stock trading scenarios involving multiple stocks.

Table 1: cumulative returns at the end of a test episode as a fraction of the initial assets

		Stock Ticker							
		AMD	BB	MKTY	DORM	RRD	ARLP	ATLC	DLPN
Model	baseline	**1.091**	0.998	**1.005**	**1.022**	0.993	0.990	**1.027**	0.968
	DQN	1.021	1.000	**1.005**	1.020	0.999	**1.009**	1.022	0.998
	UA-DQN 0.5	1.023	1.000	1.001	1.013	1.002	1.003	1.026	1.018
	UA-DQN 1.0	1.046	**1.001**	**1.005**	1.016	1.000	1.001	1.005	**1.024**
	UA-DQN 2.0	**1.091**	**1.001**	**1.005**	1.011	**1.004**	1.004	1.025	1.013

References

[1] W. R. Clements et al. "Estimating Risk and Uncertainty in Deep Reinforcement Learning". In: *ICML* (2020).

[2] W. Dabney et al. "Distributional Reinforcement Learning with Quantile Regression". In: *AAAI Conference on Artificial Intelligence* (2018).

[3] F. Leno et al. "Uncertainty-Aware Action Advising for Deep Reinforcement Learning Agents". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence* (2020).

[4] X. Liu et al. *FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance.* 2020.

[5] B. O'Donoghue et al. "The Uncertainty Bellman Equation and Exploration". In: *Proceedings of the Thirty-Fifth International Conference on Machine Learning* (2018).

[6] A. Tamar, D. Di Castro, and S. Mannor. "Policy Gradients with Variance Related Risk Criteria". In: *Proceedings of the Twenty-Ninth International Conference on Machine Learning* (2012), pp. 387–396.

[7] Cedric Zhuang. *Stock Statistics/Indicators Calculation Helper*. 2016. URL: `https://pypi.org/project/stockstats/`.