A Review of the QLBS Model: DP and RL Solutions

Jifan He

New York University

May 2023

Content

- What is the QLBS Model?
- Comparisons with Classical BS Model
- The QLBS Model: Mathematical Rationale
- Dynamic Programming (DP): Transition Probabilities and Rewards are Known
- Reinforcement Learning (RL): Transition Probabilities and Rewards are Unknown
- Some Further Extensions: Inverse Reinforcement Learning (IRL) and Option Portfolios
- Conclusions

What is the QLBS Model?

- The QLBS Model, also called the Q-Learning Black Scholes Model, is a discrete-time option pricing model by analyzing the dynamic evolutions of the replicating portfolios of option pricing.
- The QLBS Model states the necessity of keeping times discrete and derives the optimal hedging and pricing both of the parts of the same optimal value Bellman equation (will be shown later).
- The QLBS Model can be extended to other financial applications as well, such as pricing exotic options and even the portfolios of options, which will be discussed later in this presentation.

Comparisons with Classical BS Model

Similarities

- Both are methods used to price options
- Both use the dynamic hedging portfolio approach to derive the option pricing formula
- When $\Delta t \to 0$, and the stock prices follow the lognormal distributions, QLBS Model's formulas are the same as the ones of the BS Model

Differences

- The QLBS Model states the significance of the existence of options, while BS Model does not
- The QLBS Model is more practical for real-world applications: it treats time steps as discrete
- The QLBS Model can even learn from the relatively noisy real-world applications given enough dataset
- The BS Model incurs the infamous volatility smile issue, while the QLBS Model solves it

The QLBS Model: Mathematical Rationale

Basic Ideas

- Selling a European put option with maturity T and the terminal payoff of $H_T(S_T)$ at maturity
- Dynamically hedging the short position: $\Pi_t = u_t S_t + B_t$, with $u_T = 0$: $\Pi_t = B_t = H_T(S_T)$
- Instead of asking the fair option price, *adding the cumulative expected discounted variance of the hedge portfolio along all time steps $t = 0, ..., N$, with a risk-aversion parameter λ*

Risk-Adjusted Option Price

$$C_0^{(ask)}(S, u) = \mathbb{E}_0[\Pi_0 + \lambda \sum_{t=0}^{T} e^{-rt} Var[\Pi_t | F_t] | S_0 = S, u_0 = u] \quad (1)$$

We want to minimize Eq.(1) to make the put option as competitive as possible. But this minimization problem is equivalent to the maximization problem of the negative of (1)!

Minimization of a fair option price: max $(V_t = -C_t^{(ask)})$

$$V_t(S_t) = V_t^{\pi}(X_t) = \mathbb{E}_t[-\Pi_t - \lambda \sum_{t'=t}^{T} e^{-r(t'-t)} Var[\Pi_{t'}|F_{t'}]|F_t] \quad (2)$$

Bellman Equation

$$V_t^{\pi}(X_t) = \mathbb{E}_t[R(X_t, a_t, X_{t+1}) + \gamma V_{t+1}^{\pi}(X_{t+1})] \quad (3)$$

Notes

- $R_t(X_t, a_t, X_{t+1}) = \gamma a_t \Delta S_t(X_t, X_{t+1}) - \lambda Var[\Pi_t|F_t] =$
  $\gamma a_t \Delta S_t(X_t, X_{t+1}) - \lambda \gamma^2 \mathbb{E}_t[\hat{\Pi}_{t+1}^2 - 2a_t \Delta \hat{S}_t \hat{\Pi}_{t+1} + a_t^2 (\Delta \hat{S}_t)^2] \quad (4)$

Action-Value Function: Q Function

The action-value function, or Q-function, uses the same expression as Eq.(3), but it is conditioned on both the current state X_t and the initial action $a = a_t$, following a policy π afterward:

- $Q_t^{\pi}(x, a) = \mathbb{E}_t^{\pi}[-\Pi_t - \lambda \sum_{t'=t}^{T} e^{-r(t'-t)} Var[\Pi_{t'}|F_{t'}]|X_t = x, a_t = a]$ (5)

Optimal Policy $\pi_t^*(\cdot|X_t)$

The optimal policy $\pi_t^*(\cdot|X_t)$ is the policy maximizing the value function $V_t^{\pi}(X_t)$, or maximizing the action-value function $Q_t^{\pi}(X_t, a_t)$:

- $\pi_t^*(X_t) = argmax_{\pi} V_t^{\pi}(X_t) = argmax_{a_t \in A} Q_t^*(X_t, a_t)$ (6)

The Ultimate Bellman Optimal Equation

$Q_t^{\star}(x, a) = \mathbb{E}_t\left[R_t(X_t, a_t, X_{t+1}) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^{\star}(X_{t+1}, a_{t+1}) | X_t = x, a_t = a\right]$ (7), for all $t = 0, ..., T-1$

- Terminal condition: $Q_T^*(X_t, a_T = 0) = -\Pi_T(X_T) - \lambda Var[\Pi_T(X_T)]$ for $t = T$

The Optimal Action: $a_t^*(S_t)$

The optimal action a_t^* is derived analytically to maximize the quadratic representation of Eq.(7): $a_t^*(X_t) = \frac{\mathbb{E}_t[\Delta\hat{S}_t\hat{\Pi}_{t+1}]}{\mathbb{E}_t[(\Delta\hat{S}_t)^2]}$ (8)

Optimal Action-Value Function

Plugging Eq.(8) into the quadratic representation of Eq.(7), we obtain the optimal action-value function:

- $Q_t^*(X_t, a_t^*) = \gamma\mathbb{E}_t[Q_{t+1}^*(X_{t+1}, a_{t+1}^*) - \lambda\gamma\hat{\Pi}_{t+1}^2 + \lambda\gamma(a_t^*(X_t))^2(\Delta\hat{S}_t)^2]$ (9)

QLBS Option Price

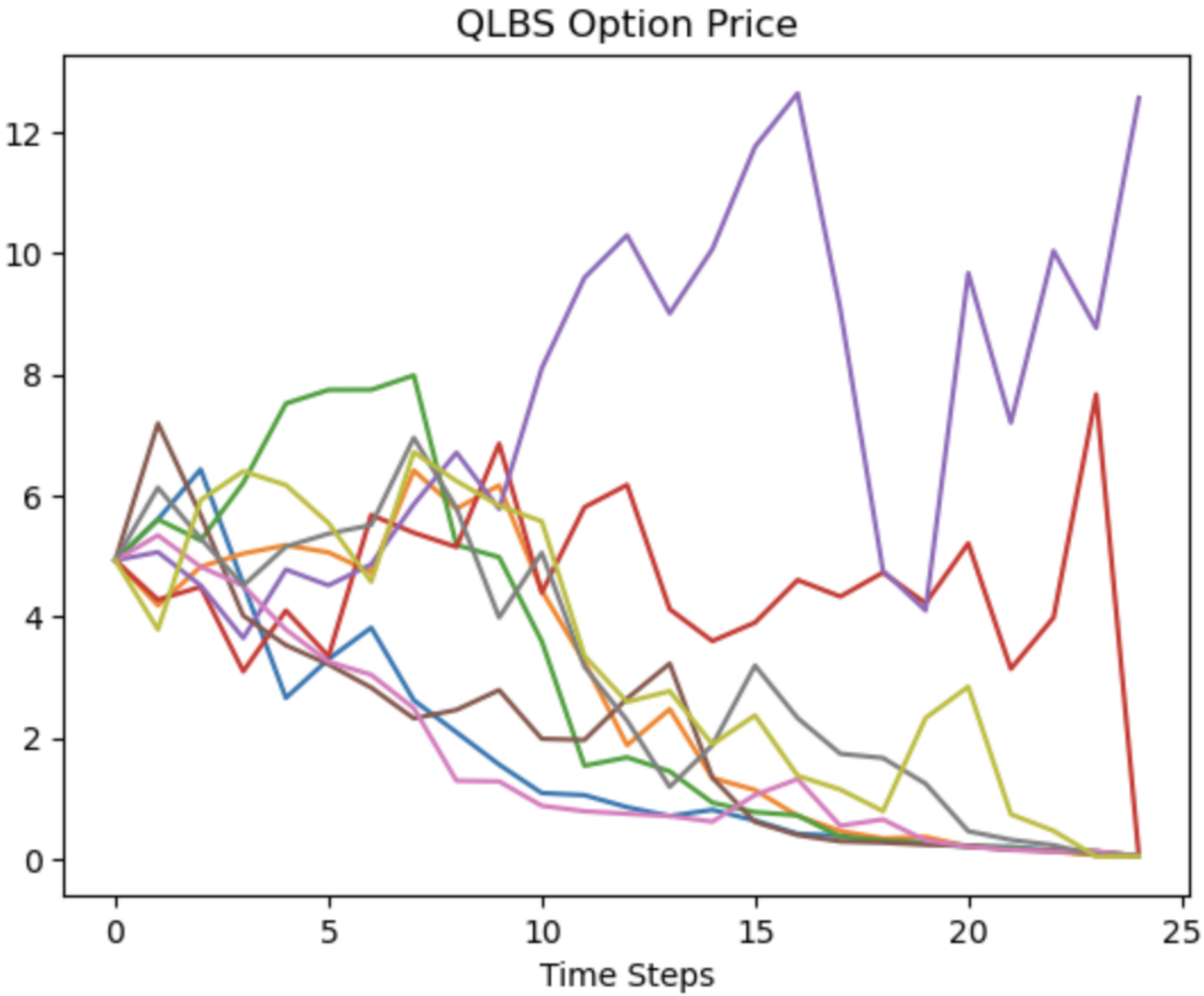
The QLBS version of option price is: $C_t^{(QLBS)}(S_t, ask) = -Q_t(S_t, a_t^*)$ (10)

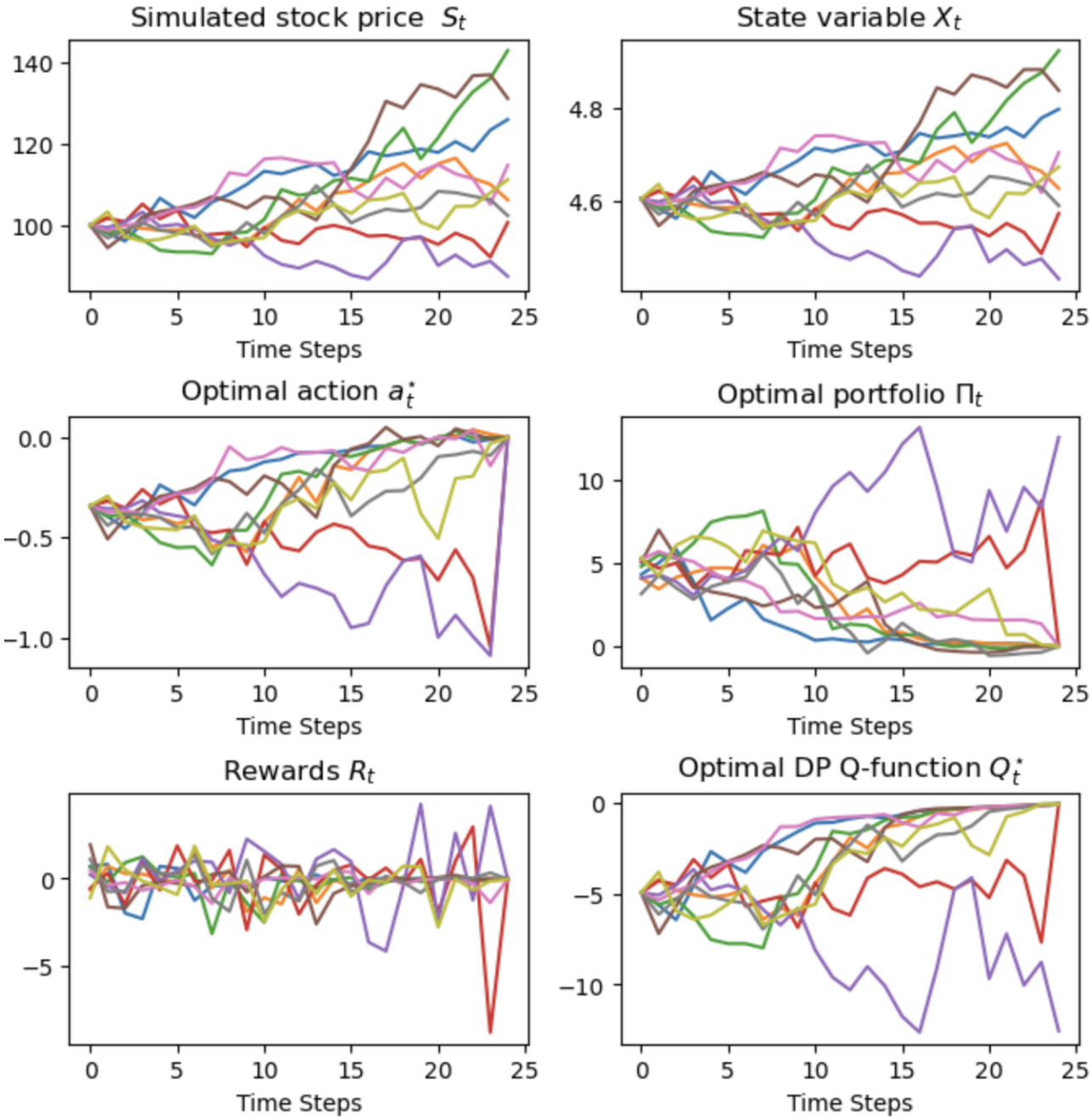
Dynamic Programming (DP): Transition Probabilities and Rewards are Known

- $a_t^\star(X_t) = \sum_n^N \phi_{nt} \Phi_n(X_t)$
- $Q_t^\star(X_t, a_t^\star) = \sum_n^N \omega_{nt} \Phi_n(X_t)$
- Coefficients ϕ_{nt}, ω_{nt} are computed recursively backward in time for $t = T-1, ..., 0$.
- $\phi_t = \mathbf{A}_t^{-1}\mathbf{B}_t$; $\omega_t = \mathbf{C}_t^{-1}\mathbf{D}_t$

Solutions – Python Codes

- $A_{nm}^{(t)} = \sum_{k=1}^{N_{MC}} \Phi_n(X_t^k) \Phi_m(X_t^k) \left(\Delta \hat{S}_t^k\right)^2$,
- $B_n^{(t)} = \sum_{k=1}^{N_{MC}} \Phi_n(X_t^k) \left[\hat{\Pi}_{t+1}^k \Delta \hat{S}_t^k + \frac{1}{2\gamma\lambda} \Delta S_t^k\right]$
- $C_{nm}^{(t)} = \sum_{k=1}^{N_{MC}} \Phi_n(X_t^k) \Phi_m(X_t^k)$,
- $D_n^{(t)} = \sum_{k=1}^{N_{MC}} \Phi_n(X_t^k) \left(R_t(X_t, a_t^\star, X_{t+1}) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^\star(X_{t+1}, a_{t+1})\right)$


QLBS Option Price
12
10
8
6
4
2
0
0
5
10
15
20
25
Time Steps

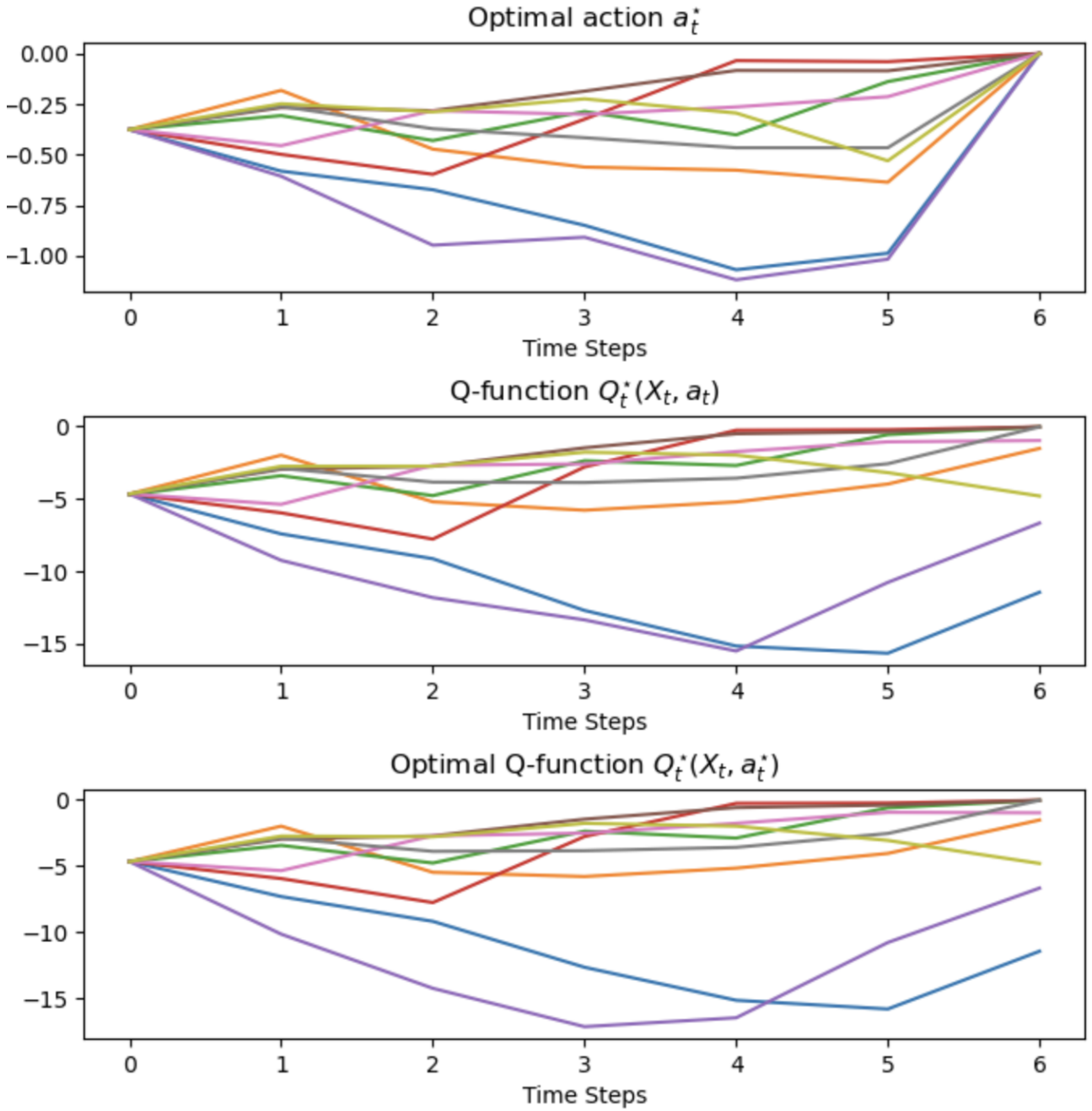

Simulated stock price S_t
140
120
100
0
5
10
15
20
25
Time Steps
State variable X_t
4.8
4.6
0
5
10
15
20
25
Time Steps
Optimal action $a_t^\star$
0.0
−0.5
−1.0
0
5
10
15
20
25
Time Steps
Optimal portfolio Π_t
10
5
0
0
5
10
15
20
25
Time Steps
Rewards R_t
0
−5
0
5
10
15
20
25
Time Steps
Optimal DP Q-function $Q_t^\star$
0
−5
−10
0
5
10
15
20
25
Time Steps

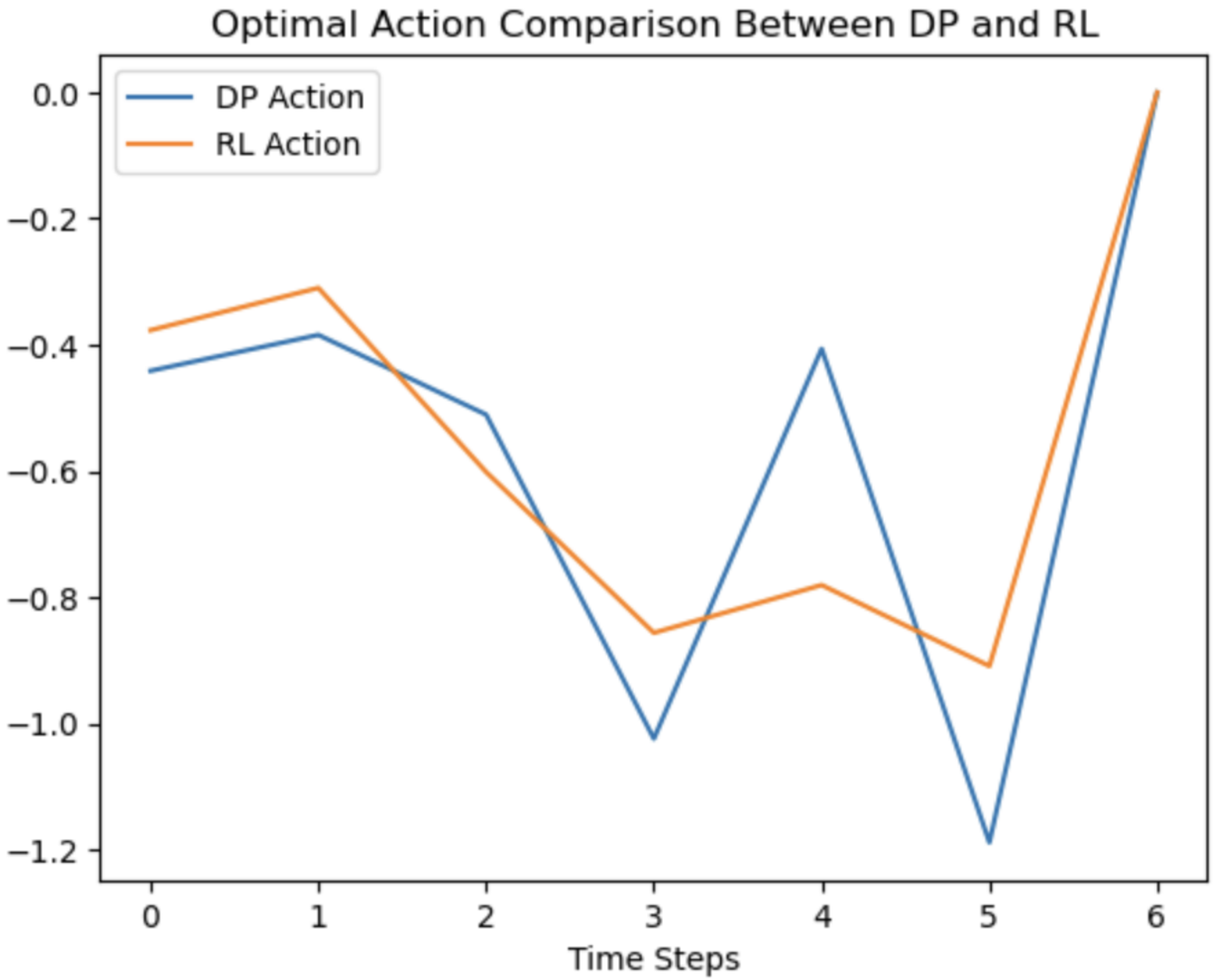
Reinforcement Learning (RL): Transition Probabilities and Rewards are Unknown

- The off-policy data, random action and the corresponding reward, is used in the RL method in this situation
- This is done by disturbing the optimal action $a_t^*(X_t)$ by (multiplying) a random noise $a_t(X_t) = a_t^\star(X_t) \sim U[1-\eta, 1+\eta]$
- $Q_t^*(X_t, a_t) = \mathbf{A}_t^T \mathbf{W}_t \Phi(X) = \sum_{i=1}^{3} \sum_{j=1}^{M} (\mathbf{W}_t \odot (\mathbf{A}_t \otimes \Phi^T(X)))_{ij} = \vec{W}_t \cdot vec(\mathbf{A}_t \otimes \Phi^T(X)) \equiv \vec{W}_t \vec{\Psi}(X_t, a_t)$
- $\vec{W}_t^* = \mathbf{S}_t^{-1} \mathbf{M}_t \rightarrow$ *Fitted Q Iteration (FQL) model*

Solutions – Python Codes

- $S_{nm}^{(t)} = \sum_{k=1}^{N_{MC}} \Psi_n\left(X_t^k, a_t^k\right) \Psi_m\left(X_t^k, a_t^k\right)$,
- $M_n^{(t)} = \sum_{k=1}^{N_{MC}} \Psi_n\left(X_t^k, a_t^k\right) \left(R_t\left(X_t, a_t, X_{t+1}\right) + \gamma \max_{a_{t+1} \in \mathcal{A}} Q_{t+1}^\star\left(X_{t+1}, a_{t+1}\right)\right)$


Optimal action a_t^*
0.00
−0.25
−0.50
−0.75
−1.00
0
1
2
3
4
5
6
Time Steps
Q-function $Q_t^*(X_t, a_t)$
0
−5
−10
−15
0
1
2
3
4
5
6
Time Steps
Optimal Q-function $Q_t^*(X_t, a_t^*)$
0
−5
−10
−15
0
1
2
3
4
5
6
Time Steps


Optimal Action Comparison Between DP and RL
DP Action
RL Action
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
0
1
2
3
4
5
6
Time Steps

Some Further Extensions: Inverse Reinforcement Learning (IRL) and Option Portfolios

IRL

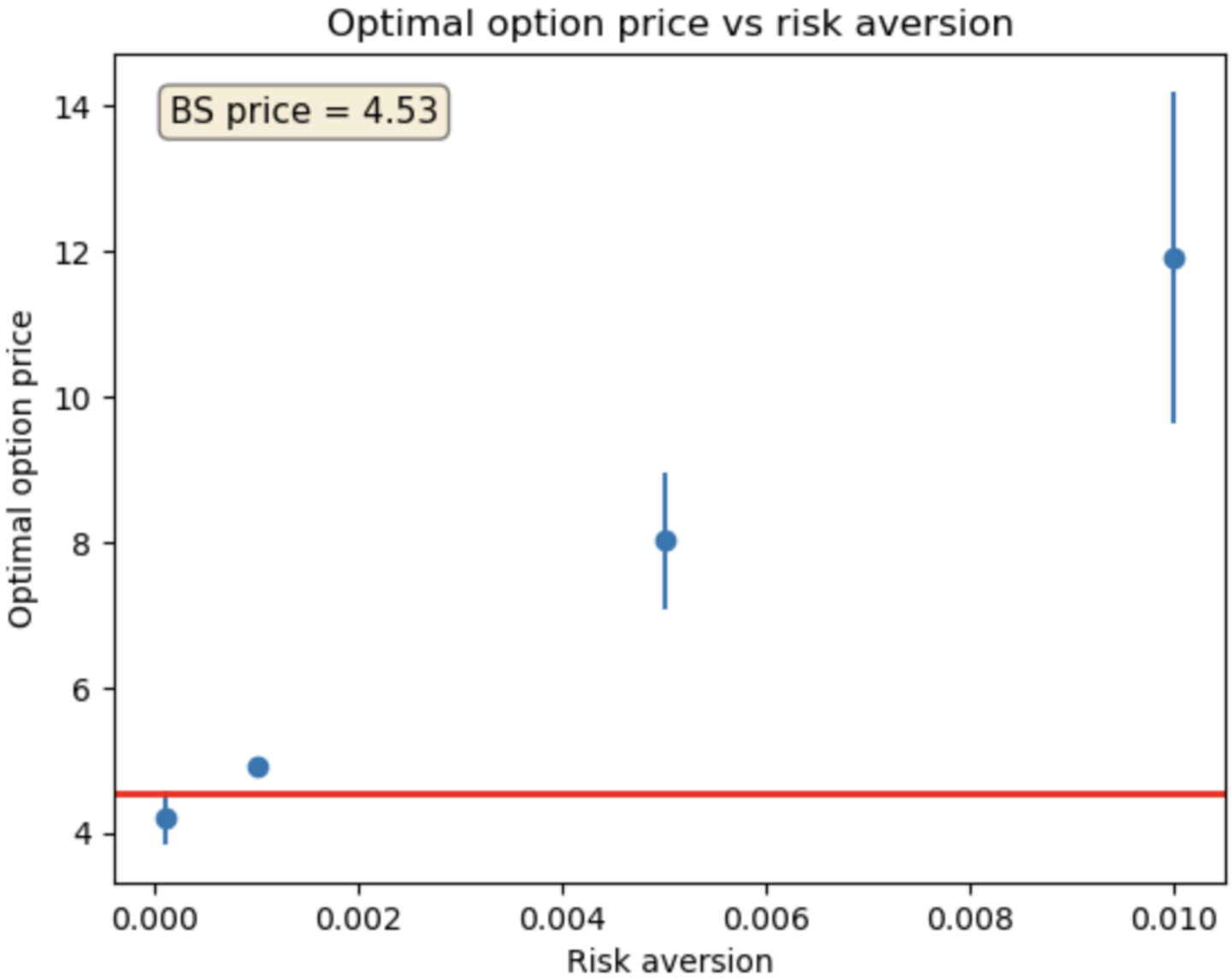
- Under IRL, there is no information on rewards
- Task for IRL: (1) find $R_t^{(n)}$ most consistent with observed states and actions; and (2) find the optimal policy and action-value function, the same as RL
- Biggest Challenge: Finding the Markowitz risk-aversion parameter λ
- Solutions: **maximum Entropy IRL** (using the G-Learning to solve the discrepancies between IRL and RL)

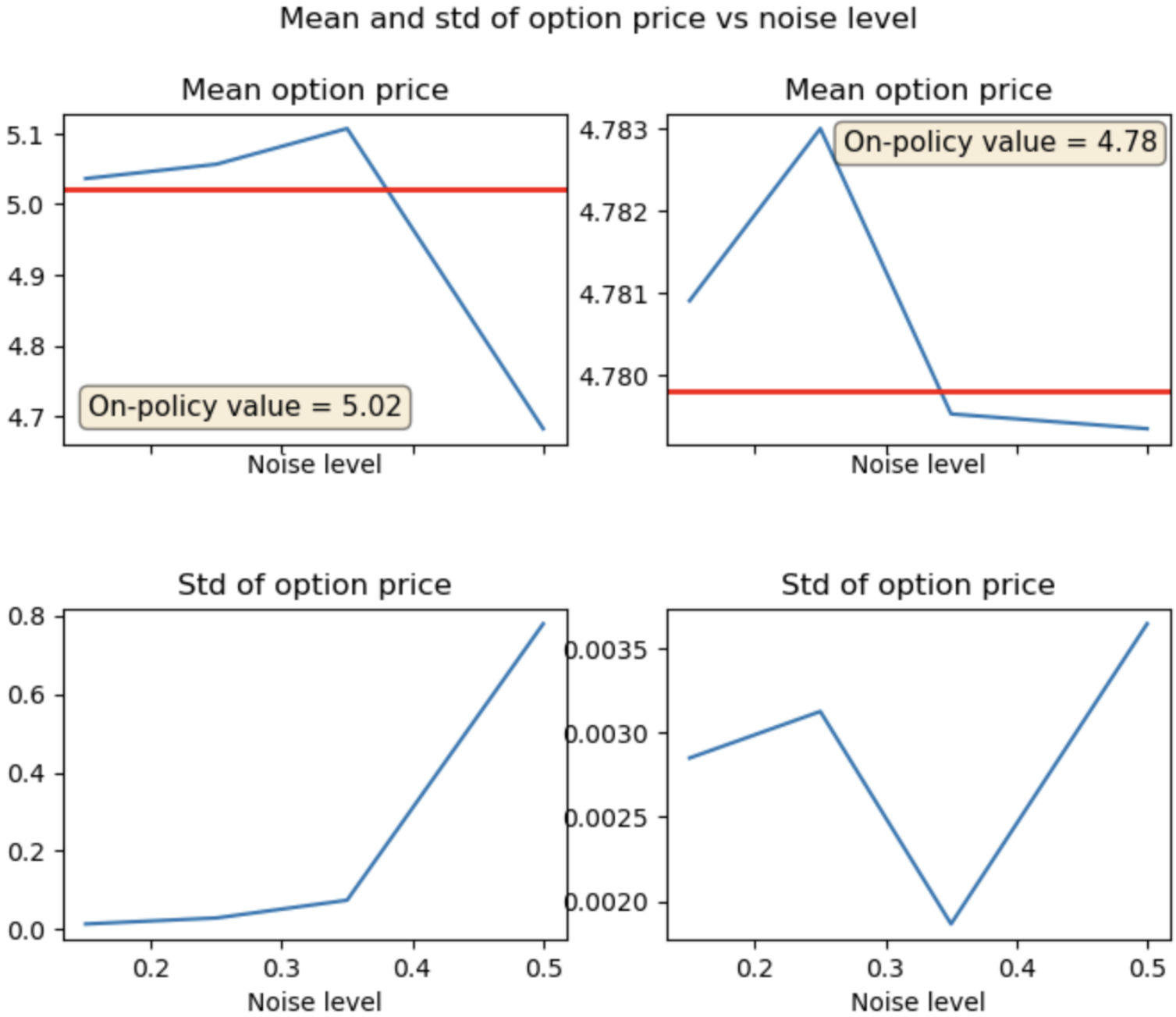
Option Portfolios

- Pricing option portfolios is the same way as pricing the option: we simply extend vectors of actions $\mathbf{a}_t^{(n)}$ and rewards $\mathbf{R}_t^{(n)}$ and use the FQL algorithm
- When adding the exotic option to an existing porfolio consisting of options, we can use a *proxy* option to extract the delta and reward information if the option was not previously traded before

Conclusions

- The key difference between the QLBS pricing model and the classical BS pricing model is that the former places main emphasis on the risk-return analysis of option replicating portfolios, while the latter does not say anything about the expected risk in option positions
- QLBS model is rooted in DP and RL, unlike the classical BS model rooted in Ito's calculus, and convergences to the BS mdoel's result under $\Delta t \to 0$, and the prices follow lognormal
- QLBS model is consistent even when pricing option portfolios and the volatility smile problem does not appear in QLBS due to its reliance on data instead of a model
- QLBS model is powerful in learning the real-world application in terms of trader's sub-optimal hedging actions and even purely randomness given enough data and advanced GPUs/TPUs


Optimal option price vs risk aversion
BS price = 4.53
Optimal option price
14
12
10
8
6
4
0.000
0.002
0.004
0.006
0.008
0.010
Risk aversion


Mean and std of option price vs noise level
Mean option price
5.1
5.0
4.9
4.8
4.7
On-policy value = 5.02
Noise level
Mean option price
4.783
4.782
4.781
4.780
On-policy value = 4.78
Noise level
Std of option price
0.8
0.6
0.4
0.2
0.0
0.2
0.3
0.4
0.5
Noise level
Std of option price
0.0035
0.0030
0.0025
0.0020
0.2
0.3
0.4
0.5
Noise level

Thank you!