Reinforcement learning for financial markets

Eric Benhamou

Université Paris Dauphine

eric.benhamou@dauphine.eu

Abstract

The main hypothesis underlying most financial markets models is market efficiency [Lo (2005)]. Market prices should fully reflect all available information and it should be impossible to systematically capture profits in financial markets.

In practice, human agents' decisions are affected by some behavior distortions like risk aversion, superstition, misperception and mis-assessment of probabilities that make the game not as efficient as thought.

In this on going research, we investigate the creation of an intelligent trading agent that adapts to the complexity of financial markets and performs profitable trading. We cast the general problem into a stochastic control problem and explores two traditional Reinforcement Learning algorithms to solve it (Q-Learning and SARSA) as in [Bertoluzzo-Corazza (2015)]. We extend their results by taking a full matrix representation. We present results that shows that RL algos outperform a simple buy and hold strategy.

1. Primer on RL

Investing in financial markets is not easy. As prices roll up and down, choosing a long or short position is a complex task, stressful and risky. It can be very emotionally and challenging. Without an edge, it is like playing Russian roulette.

Reinforcement learning (RL) is an area of machine learning inspired by behaviourist psychology, that helps reformulating the decision problem as a dynamic programming problem where an agent ought to take actions so as to maximize some notion of cumulative reward (for instance see [Sutton-Barto (2017)]). The general setting of Reinforcement learning is summarized by figure 1 below

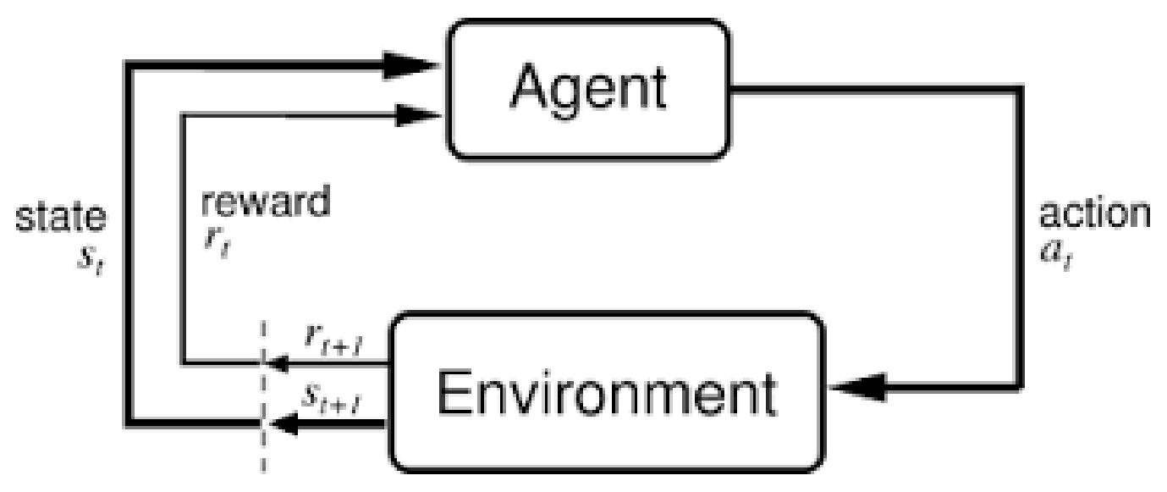

Figure 1: Classical Agent-Environment interaction

At each discrete time step, $t = 0, 1, 2..., N$, the agent:
- receives a reward $r_t \in \mathbb{R}$ as a response to its previous actions a_{t-1}
- perceives the current state $s_t \in \mathcal{S}$ of the environment
- selects an action $a_t \in \mathcal{A}(s_t)$ where $\mathcal{A}(s_t)$ is the set of available actions in state s_t

The agent's objective is to maximize the sum of its total discounted rewards denoted by R_t given by $R_t = \sum_{k=0}^{+\infty} \gamma^k r_{t+k+1}$, where $\gamma \in [0, 1]$ is the discount rate.

This problem is classically reformulated in terms of a Q matrix as follows. At each step t, the agent, starting from the state s_t, has to calculate the value of the action-value function $Q^\pi(s, a)$ for each possible action on the basis of the policy π :

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[R_t | s_t = s, a_t = a \right]$$

A fundamental property of the action-value function is the fulfillment of the following special recursive relationship, known as Bellman equation for $Q^\pi(s, a)$:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) | s_t = s, a_t = a \right]$$

This Bellman equation has a unique solution that can be computed numerically with various algorithms.

2. Q-Learning & SARSA algorithms

Two traditional algorithms to solve the Bellman equations are Q-learning and SARSA . They compute recursively the Q matrix thanks to a stochastic graident approach:

$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + \alpha \times Step$$

where $\alpha \in [0, 1]$ is the learning rate and the step is in a sense a sort of gradient that differs in the two methods.
In Q learning, we take the "optimistic" view that all future action selections from every state should be optimal, thus we pick the action a that maximizes $Q(s_{t+1}, a)$. This leads to an optimal state given by the next step reward and the optimal decision:
$Step = r_{t+1} + \gamma \max_a Q_k(s_{t+1}, a_{t+1}) - Q_k(s_t, a_t)$

In SARSA, the agent learns $Q^*(s, a)$ on the basis of the performed action, hence the step is simpler to compute and guven by
$Step = r_{t+1} + \gamma Q_k(s_{t+1}, a_{t+1}) - Q_k(s_t, a_t)$

In the trading environment, one has to define the world and the reward. We follow [Bertoluzzo-Corazza (2015)] and take the last $N = 5$ logaritmic returns. The reward is the traditional Sharpe ratio over the last 20 days (corresponding to a full month). we take a greedy step $\varepsilon = 5\%$ and transaction brokerage cost equal to 0.20% of the stock value. We take as a benchmark a linearized approximation of the Q value function parametrized by θ as follows

$$Q_{\theta_k}(s_t, a_t) = \theta_{k,0} + \textstyle\sum_{n=1}^{N+1} \theta_{k,n}\phi(s_{t,n}) + \theta_{k,N+2}\phi(a_t)$$

where $\phi(.)$ is a logistic function. Our improvement is to realize that we can also take what we call a 'full matrix' approach and store all Q matrix coefficients given by $Q(s_t, a_t)$.

As s_t takes $3^5 = 243$ possible values, and a_t takes 3 (short, neutral, long), we have a 243 rows by 3 columns matrix. This is largely manageable. We show that this provides better results as there is no more approximation of the Q matrix.

3. Results and Discussion

To compare with [Bertoluzzo-Corazza (2015)], we investigate the value of Telecom Italia for a period from January 1986 to December 2017. Data are from yahoo finance.
The results are summarized in table 1. We can see that the full matrix approach is superior. For reference, a buy and hold strategy leads to a performance of -6.9%. This shows that reinforcement learning improves substantially decision.

Method	Annual Return	Trades/year	Drawdown
Sarsa	2.23 %	2.19	-50 %
Full Sarsa	8.52 %	10.6	-47 %
Q Learning	2.31 %	2.25	-25 %
Full Q Learning	9.76 %	10.7	-35 %

Table 1: Results for the various models

On the graphics below, we can notice various facts. First, full matrix representation improves results. Second, it leads to generate more trades. This emphasizes that in this method, it is critical to keep transaction cost at the minimum. Third, this comes at the price of more drawdown. If performance is measured by recovery factor (the performance divided by the drawdown), the enhancement reduces.

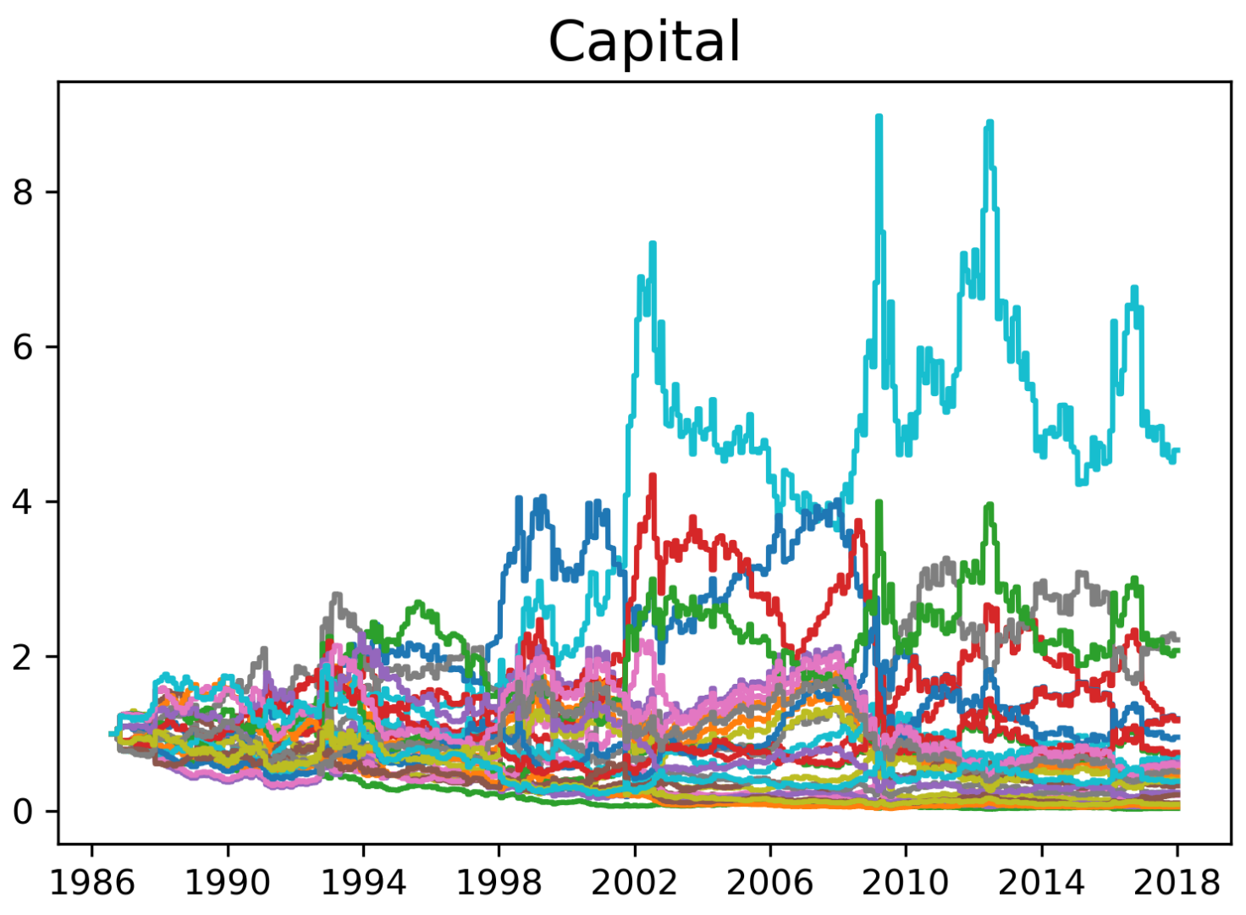

Figure 2: **Linear Q-Learning MC:** For the various Monte Carlo, simulation, we provide the equity line path for the action Telecom Italia. We can see that we have some path ending with negative performance but the majority of the paths are with positive performance

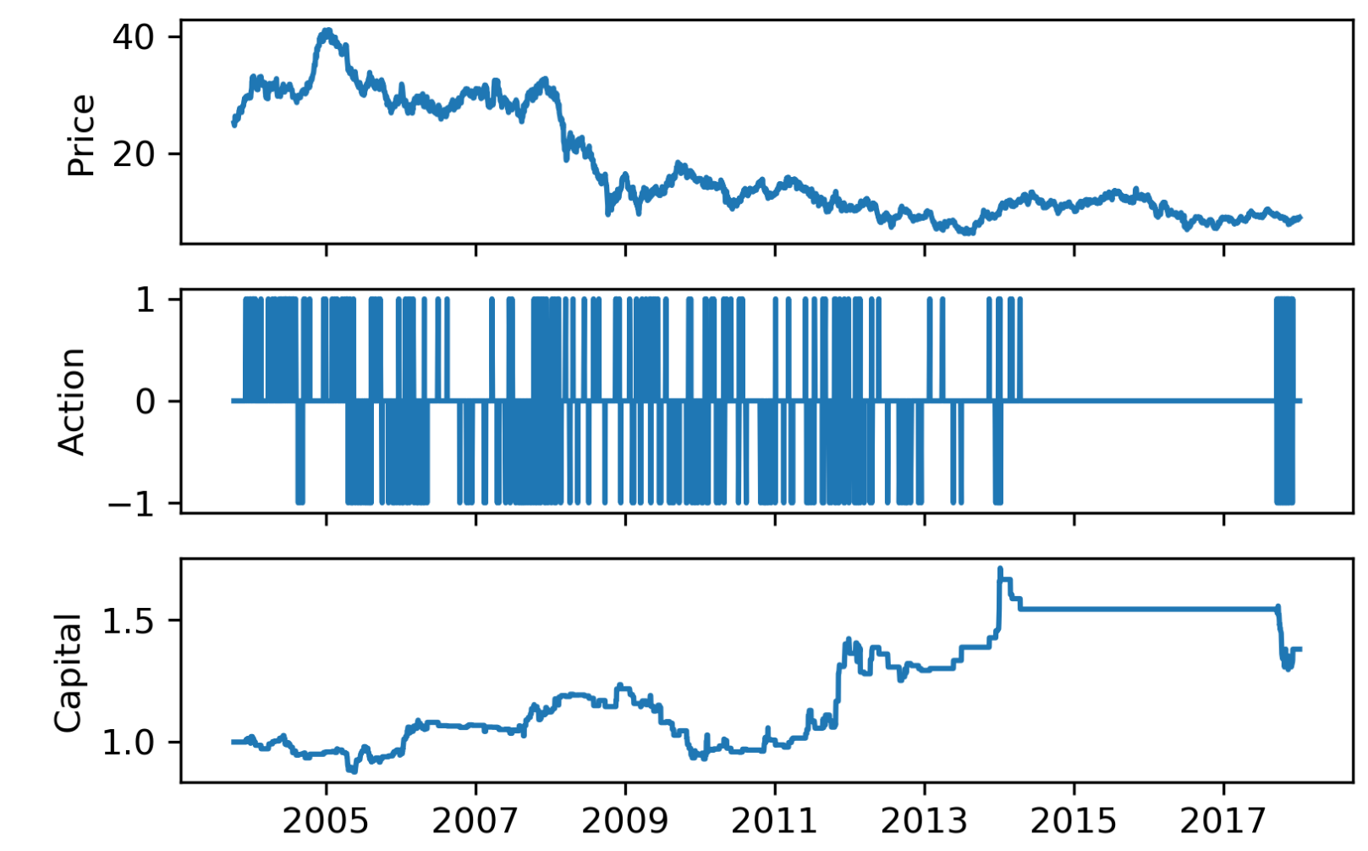

Figure 3: **Linear Q-Learning Results:** The graphics above provides the prices of Telecom Italia, the optimal actions and the resulting equity. The method is profitable as the equity line regularly increases. The trading system obtained trades only occasionally compared to the full matrix approach.

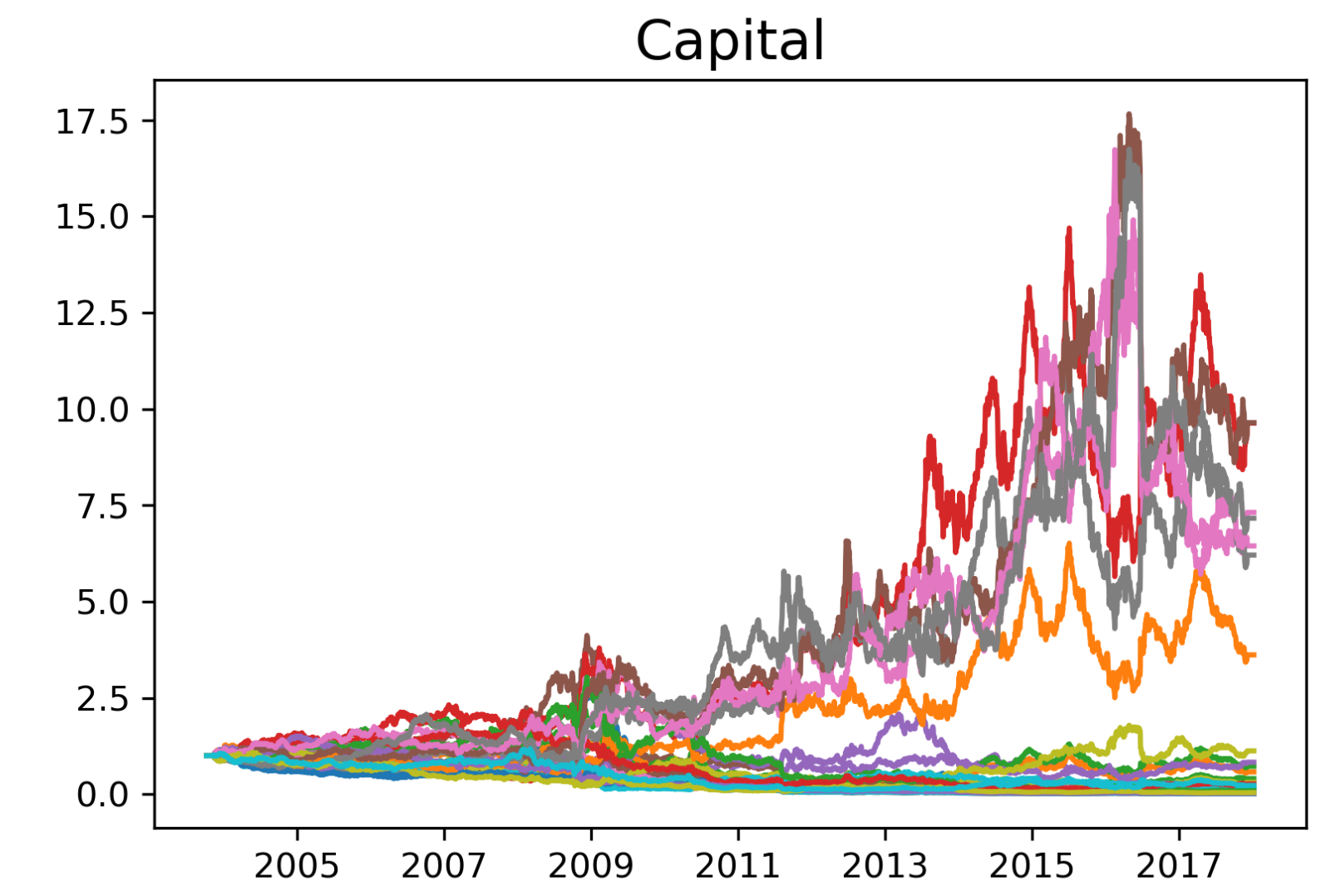

Figure 4: **Full Matrix Q-Learning MC**: Same graph as for the Q linear approximation method but using full matrix one. Note the full matrix provides higher Monte Carlo path, indicating a more efficient method

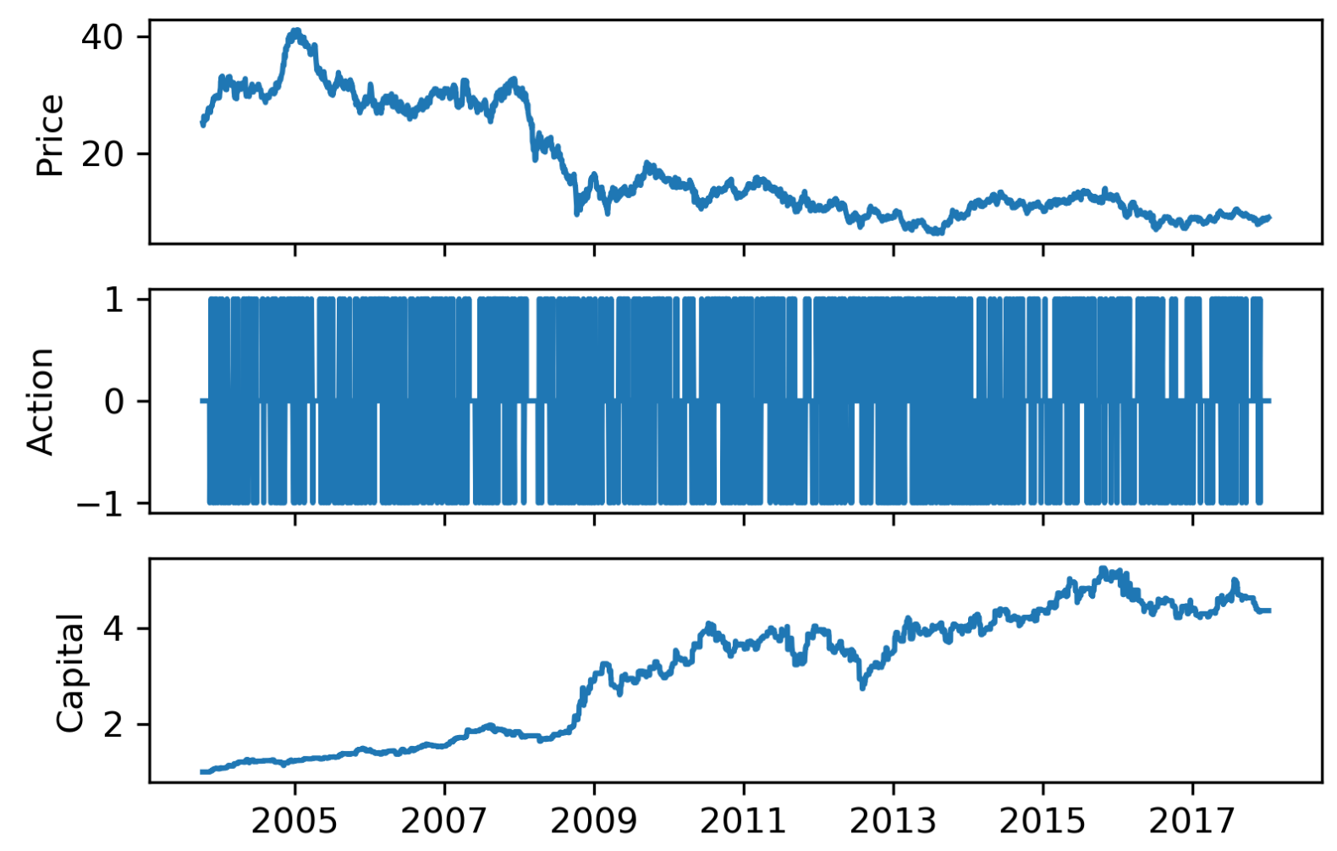

Figure 5: **Full Matrix Q-Learning Results:** Compared to linearized QLearning, the full matrix method gives higher results (average annual return of 9.76 percent compared to 2.31 percent for the linearized Q. It also generates more trades as it is more responsive to states environment

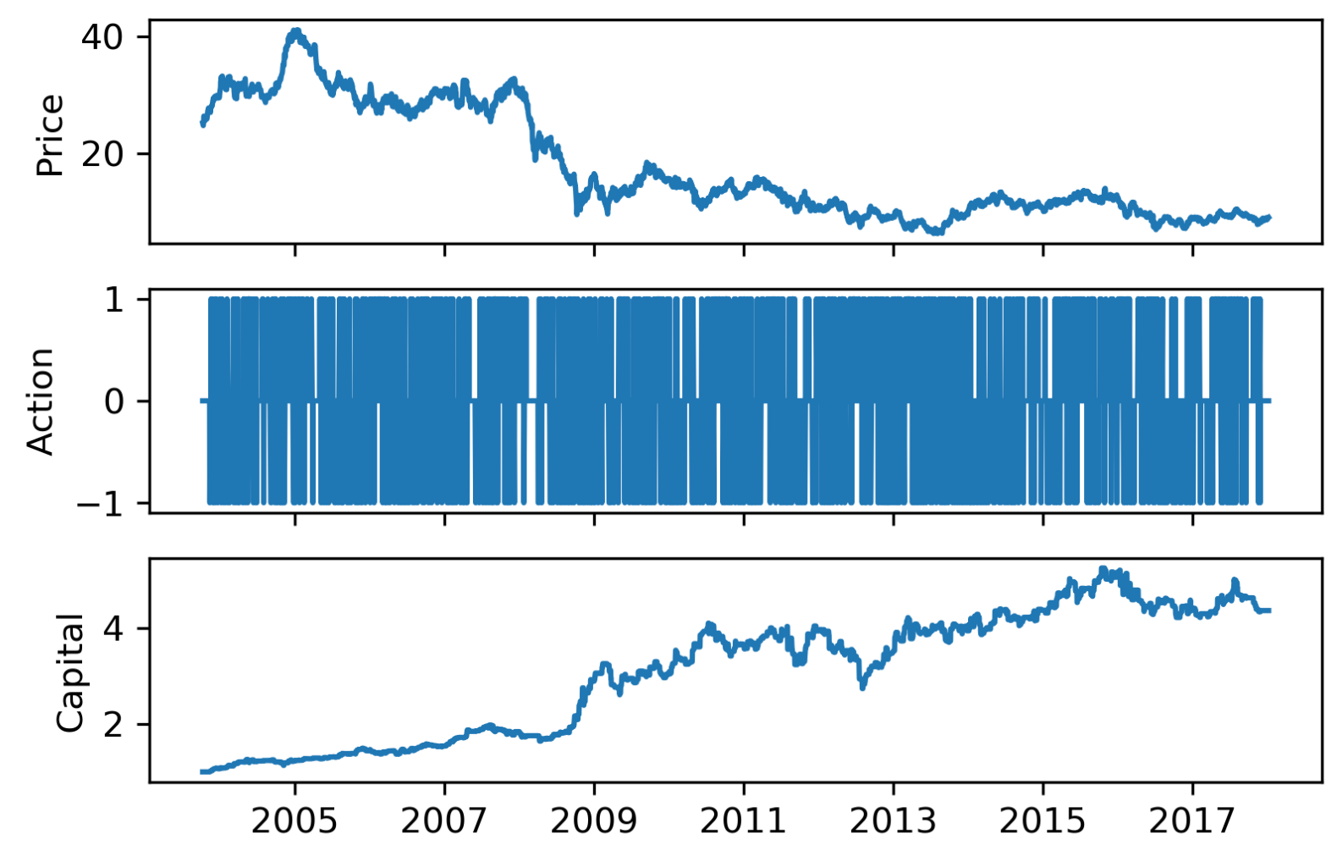

Figure 6: **Full Matrix SARSA Results:** This is the graphics for the SARSA method using full matrix approach. Like the Q-Learning case, the full matrix approach generates more trades and better results (8.52 percent versus 2.23)

Python source code and data for the experiment can be found on github [Benhamou (2018)]

4. Conclusion

Our improvement to compute the full matrix leads to better performance both for Q and Sarsa learning. This improvement comes at the price of more frequent trades and hence larger drawdowns that makes the enhancement questionable in practice. In further reasearch, we would like to add additional states to capture situtations where it is not optimal to trade because of past performance.

References

[Benhamou (2018)] E. Benhamou Reinforcement learning for Optimal trading, Github https://github.com/ericbenhamou/RL, 2018

[Bertoluzzo-Corazza (2015)] F. Bertoluzzo and M.Corazza: Q-Learning and SARSA: A Comparison between Two Intelligent Stochastic Control Approaches for Financial Trading, SSRN 2015

[Lo (2005)] A.W. Lo: Reconciling efficient markets with behavioral finance: The Adaptive Markets Hypothesis. The Journal of Investment Consulting, 7, 21-44, 2005.

[Sutton-Barto (2017)] R.S. Sutton and A.G. Barto: Reinforcement Learning. An Introduction. Second Edition, in progress MIT Press, Cambridge, MA, 2017