# Metadata of the chapter that will be visualized in SpringerLink

| | | |
|---|---|---|
| Book Title | Mathematical and Statistical Methods for Actuarial Sciences and Finance | |
| Series Title | | |
| Chapter Title | Comparing RL Approaches for Applications to Financial Trading Systems | |
| Copyright Year | 2021 | |
| Copyright HolderName | The Author(s), under exclusive license to Springer Nature Switzerland AG | |



| | | |
|---|---|---|
| Corresponding Author | Family Name | **Corazza** |
| | Particle | |
| | Given Name | **Marco** |
| | Prefix | |
| | Suffix | |
| | Role | |
| | Division | Department of Economics |
| | Organization | Ca’ Foscari University of Venice |
| | Address | Sestiere Cannaregio 873, Venice, Italy |
| | Email | corazza@unive.it |
| Author | Family Name | **Fasano** |
| | Particle | |
| | Given Name | **Giovanni** |
| | Prefix | |
| | Suffix | |
| | Role | |
| | Division | Department of Management |
| | Organization | Ca’ Foscari University of Venice |
| | Address | Sestiere Cannaregio 873, Venice, Italy |
| | Email | fasano@unive.it |
| Author | Family Name | **Gusso** |
| | Particle | |
| | Given Name | **Riccardo** |
| | Prefix | |
| | Suffix | |
| | Role | |
| | Division | Department of Economics |
| | Organization | Ca’ Foscari University of Venice |
| | Address | Sestiere Cannaregio 873, Venice, Italy |
| | Email | rgusso@unive.it |
| Author | Family Name | **Pesenti** |
| | Particle | |
| | Given Name | **Raffaele** |
| | Prefix | |
| | Suffix | |

| | |
|---|---|
| Role | |
| Division | Department of Management |
| Organization | Ca’ Foscari University of Venice |
| Address | Sestiere Cannaregio 873, Venice, Italy |
| Email | pesenti@unive.it |

| | |
|---|---|
| Abstract | In this paper we present and implement different Reinforcement Learning (RL) algorithms in financial trading systems. RL-based approaches aim to find an optimal policy, that is an optimal mapping between the variables describing an environment state and the actions available to an agent, by interacting with the environment itself in order to maximize a cumulative return. In particular, we compare the results obtained considering different on-policy (SARSA) and off-policy (Q-Learning, Greedy-GQ) RL algorithms applied to daily trading in the Italian stock market. We both consider computational issues and investigate practical applications, in an effort to improve previous results while keeping a simple and understandable structure of the used models. |

| | |
|---|---|
| Keywords (separated by '-') | [redacted] |

# Comparing RL Approaches for Applications to Financial Trading Systems

**Marco Corazza, Giovanni Fasano, Riccardo Gusso, and Raffaele Pesenti**

**Abstract** In this paper we present and implement different Reinforcement Learning (RL) algorithms in financial trading systems. RL-based approaches aim to find an optimal policy, that is an optimal mapping between the variables describing an environment state and the actions available to an agent, by interacting with the environment itself in order to maximize a cumulative return. In particular, we compare the results obtained considering different on-policy (SARSA) and off-policy (Q-Learning, Greedy-GQ) RL algorithms applied to daily trading in the Italian stock market. We both consider computational issues and investigate practical applications, in an effort to improve previous results while keeping a simple and understandable structure of the used models.

**Keywords** ■■■

## 1 Introduction

In this paper, we propose some automated Financial Trading Systems (FTSs) based on a self-adaptive machine learning approach known as Reinforcement Learning (RL). Specifically, we define our FTSs on the basis of the following RL methodologies:



M. Corazza (✉) · R. Gusso
Department of Economics, Ca' Foscari University of Venice, Sestiere Cannaregio 873, Venice, Italy
e-mail: corazza@unive.it

R. Gusso
e-mail: rgusso@unive.it

G. Fasano · R. Pesenti
Department of Management, Ca' Foscari University of Venice, Sestiere Cannaregio 873, Venice, Italy
e-mail: fasano@unive.it

R. Pesenti
e-mail: pesenti@unive.it

 
M. Corazza et al. (eds.), *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, https://doi.org/10.1007/978-3-030-78965-7_22

*State-Action-Reward-State-Action* (SARSA) [1, 9] and *Q-Learning* (QL) [1, 10], with its development *Greedy-GQ* [8]. Then, we compare their effectiveness.

The considered methodologies concern an agent interacting with an environment. The agent perceives the state of the environment and takes an action, then the environment provides a negative or a positive reward to the action. This iterative process allows the agent to heuristically identify a policy that maximizes a cumulative return over time. In our case, the agent is a FTS, the environment is a financial market and the reward is a measure of financial gain/loss. The FTS has to decide a trading strategy, i.e., when to sell or to buy an asset, or to stay out of the market. Note that the knowledge of a given FTS is not acquired in some preliminary in-sample training phase. Indeed, any action is taken by the considered FTS on the ground of the "experience" it gained up to that moment through a trial-and-error mechanism based on the rewards it obtained as consequences of its past actions.

The application of the above methodologies is justified in the assumption that the Adaptive Market Hypothesis (AMH) [7] holds. Under this perspective, a financial market can be viewed as an evolutionary environment in which different partly rational "species" (e.g., hedge funds, retail investors and others) interact among them in order to achieve a satisfactory, not necessarily optimal, level of profitability. The adaptations of these species to the various stimuli is neither instantaneous nor immediately appropriate, and this generally does not imply the efficiency of the financial market. Within this framework, a FTS agent can be seen as possibly able to learn the time-varying dynamics of the financial market, aiming at defining a profitable financial trading policy. Note that SARSA, QL and Greedy-GQ methodologies are heuristics that cannot guarantee of providing optimal solutions. On the other hand, they can be successfully applied when there is no a-priori knowledge of the transition probability matrices of the state of a dynamic environment [6, p. 199] as in the case of the financial market.

The remainder of the paper is organized as follows. In the next section, we describe the background of RL theory. In Sect. 3 we introduce our implementations of the FTSs and consider the problem of the description of the financial environment state. In Sect. 4 we analyze the results obtained by applying the developed FTSs to some stocks of the Italian FTSE Mib market.

## 2 RL Background

RL applies to problems where the following elements can be identified: (i) the *agent*, which is a learning decision maker; (ii) the *environment* the agent interacts with, in subsequent time steps; (iii) a set of possible *actions* to choose among at each time step; (iv) a feedback signal, the *reward*, from the environment.

Let us denote by $\mathscr{S}$, $\mathscr{A}$ and $\mathscr{R}$ respectively the sets of all possible states of the environment, actions and rewards. At each time step $t$ the agent reads a description of the environment current *state* $S_t \in \mathscr{S}$ and selects an *action* $A_t \in \mathscr{A}$, among the possible ones at the current state. At the subsequent time step $t + 1$, the agent receives

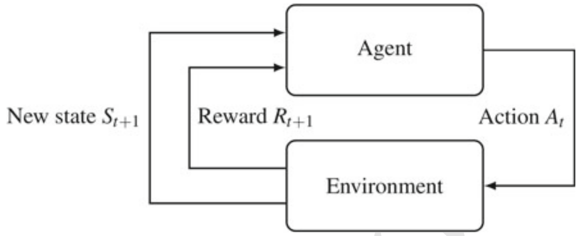


**Fig. 1** Interaction between agent and environment at time steps $t$ and $t+1$

both a reward $R_{t+1} \in \mathscr{R}$ and the description of the new environment state $S_{t+1}$ (see Fig. 1). The next assumption holds.

**Assumption 2.1** The sets $\mathscr{S}$, $\mathscr{A}$ and $\mathscr{R}$ have a finite number of distinct elements, with $\mathscr{R} \subset \mathbb{R}$. Then, random variables $R_t$, $S_t$ have a discrete probability distribution conditioned only on preceding state and action, i.e.

$$p(s', r|s, a)\ \mathbb{P}\left[S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\right], \tag{1}$$

which expresses the so-called Markov property of the state.

At each time $t$, the agent's objective is to maximize the future rewards. This task is generally achieved adopting a cumulated *discounted return* with respect to discount rate $0 \leq \gamma \leq 1$, i.e.

$$G_t\ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{2}$$

To reach the above goal, at each time $t$ the agent dynamically defines and updates a policy $\pi(\alpha|\xi)$, which determines the probability for the agent to choose an action $\alpha \in \mathscr{A}(\xi)$, given a state $\xi \in \mathscr{S}$, in order to maximize the expected value of (2), i.e. maximizing

$$q_\pi(s, a)\ \mathbb{E}_\pi\left[G_t|S_t = s, A_t = a\right]. \tag{3}$$

Here the expected value $\mathbb{E}_\pi$ is meant to be computed given that the agent selects the policy $\pi$ after choosing $a \in \mathscr{A}(s)$.

An *optimal* policy $\pi^*$ such that $q_{\pi^*}(s, a) = \max_\pi q_\pi(s, a)$ can be theoretically found solving the following Bellman equation [2]:

$$q_{\pi^*}(s, a) = \sum_{s' \in \mathscr{S}} \sum_{r \in \mathscr{R}} p(s', r|s, a)\left[r + \gamma \max_{a' \in \mathscr{A}(s')} q_{\pi^*}(s', a')\right]. \tag{4}$$

In principle, Eq. (4) might be solved if the dynamic conditioned probabilities $p(s', r|s, a)$ were known. However, even if this assumption holds, computation burden often results too heavy to be implemented in the practice.

For the above reason, RL methods would rather determine sub-optimal policies, using information the agent obtains by direct interaction with the environment, with-out assuming a complete knowledge of the probabilities $p(s', r|s, a)$. Specifically, RL gets this knowledge from sample sequences of actual or simulated states, actions, and rewards. As an example, let $Q(S_t, A_t)$ be the current estimate of $q_{\pi*}(s, a)$ for encountered state $S_t$ and chosen action $A_t$ and let $R_t$ represent the computed reward at time $t$, and $\beta_t$ is a step-size parameter. Then SARSA uses the following update rule for $Q(S_t, A_t)$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \beta_t \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right]. \quad (5)$$

## 3 The FTSs

In this section we apply the three methodologies listed in Sect. 1 to the development of automated FTSs operating on Italian FTSE stock market. The source of the data we used is the Bloomberg© database [3], from which we collected daily close prices for five major companies (Enel, Generali, Intesa, Tim, Unicredit) between January 2000 to October 2018. Our aim is to improve the results obtained in [4], while keeping a similar simple structure of both the state space representing the stock market and the trading actions available.

Then we assume that at every time step $t$ the trading system can invest all of its current budget at opening or keeping a short/long position on a single stock, or it can close it and stay out of the market. This is formalized by setting $\mathscr{A}(\mathscr{S}_t) = \mathscr{A} = \{-1, 0, 1\}$ for each time $t$ and each state $S_t$. Actions are chosen according to a policy derived from the current approximation of the $q_{\pi*}(s, a)$ function for the selected methodology.

As representation of environmental state, we generalize the approach used in [4] by considering features not only for a given number $n$ of past logarithmic returns of the considered stock price, but also for the current performance of the trade in action. Formally, we first consider the vector $\mathbf{y}(S_t, A_t) \in \mathbb{R}^{n+1}$ defined by

$$y_i(S_t, A_t) = \phi\left(\ln\left(\frac{P_{t-n+i}}{P_{t-(n+1)+i}}\right)\right), \quad \text{for } i = 1, \ldots, n \quad (6)$$

$$y_{n+1}(S_t, A_t) = \phi(PL_t) \quad (7)$$

where $PL_t = 0$ if $A_{t-1} = 0$, otherwise it is the logarithmic return of the current trade, and $\phi(x)$ is the same real-valued logistic function used in [4].

Then, for the actual feature vector $\mathbf{x}(S_t, A_t)$ we adopt a block representation commonly used in RL algorithms [5]. That is, the vector $\mathbf{y}(S_t, A_t)$ is copied to one of the three slots of a zero vector with $|\mathscr{A}| \cdot (n+1) = 3 \cdot (n+1)$ elements, according to the following rule:

$$\mathbf{x}(S_t, A_t) = \begin{cases} \left[\mathbf{y}(S_t, A_t) \quad \mathbf{0}^{n+1} \quad \mathbf{0}^{n+1}\right]^\intercal, & \text{if } A_t = -1 \\ \left[\mathbf{0}^{n+1} \quad \mathbf{y}(S_t, A_t) \quad \mathbf{0}^{n+1}\right]^\intercal, & \text{if } A_t = 0 \\ \left[\mathbf{0}^{n+1} \quad \mathbf{0}^{n+1} \quad \mathbf{y}(S_t, A_t)\right]^\intercal, & \text{if } A_t = 1 \end{cases} \tag{8}$$

where $\mathbf{0}^{n+1}$ is the null vector in $\mathbb{R}^{n+1}$.

For the reward $R_{t+1}$ we considered two choices. The first one, as in [4] is

$$R_{t+1} = \frac{\mu(g_{l,t+1})}{\sigma(g_{l,t+1})} \qquad \text{(Sharpe Ratio)} \tag{9}$$

where $\mu$ and $\sigma$ are respectively the sample mean and standard deviation of the rewards calculated over the last $l$ trading days. The second one is

$$R_{t+1} = \frac{\mu(g_{l,t+1})}{1 + \max DD_{l,t+1}} \qquad \text{(Calmar Ratio)} \tag{10}$$

where $\max DD_{l,t+1}$ is the maximum drawdown, that is the difference between the maximum value of the equity gained by the trading system calculated over the last $l$ trading days and the subsequent minimum value.

## 4 Results

We considered transaction costs required for opening and closing each position, as a percentage rate of 0.15%.

We did a first analysis of the performances of the obtained FTSs by running several replications for each FTS, to compare their performance with respect to the choice of the involved step-size parameters, i.e. $\beta_t$ and some others. More specifically, we analyzed the difference in the performance between setting them constant or decreasing over time according to the required conditions to ensure the convergence of the algorithms. Indeed, it is reasonable to assume that the rewards in the stock market do not derive from a stationary probability distribution. In this case it could be argued that possibly there is not a given optimal policy. Consequently, a methodology might perform exploratory actions and learn/correct its trading-policy. So, we first considered several possible values of the step-size parameters, keeping fixed the values for $n = 5$ and $l = 5$ and we performed $N = 1000$ replications for each combination of them and each algorithm with the two reward metrics (9)–(10). Then, we selected the values of the step-size parameters that produce on average the best final equity value, and using them we performed other $N = 5000$ replications for different values of $n$ and $l$.

Generally, for each stock the annual average return (AAR) obtained by the differently set FTSs is positive. The lowest AAR is for Tim (4.28%) and the highest one is for Unicredit (79.51%). In Table 1 we show the values of the AARs, of the maximal



506948_1_En_22_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:21/7/2021 Pages: xxx Layout: T1-Standard

**Table 1** AAR, maximal drawdown (%) and Calmar Ratio for the best FTSs, and B&H AAR

| Stock | Sharpe | | | Calmar | | | Buy & hold |
|---|---|---|---|---|---|---|---|
| | Return (%) | MaxDD (%) | Calmar ratio | Return (%) | MaxDD (%) | Calmar ratio | Return (%) |
| Enel | 18.57 | 41,83 | 0.44 | 20.35 | 40.01 | 0.51 | −2.15 |
| Generali | 23.91 | 36.84 | 0.65 | 26.67 | 39.76 | 0.67 | −3.58 |
| Intesa | 54.94 | 38.22 | 1.44 | 51.49 | 43.89 | 1.17 | −3.27 |
| Tim | 32.58 | 30.82 | 1.06 | 31.27 | 36.79 | 0.85 | −11.56 |
| Unicredit | 79.51 | 42.07 | 1.89 | 76.45 | 35.38 | 2.16 | −15.43 |

**Table 2** Ratio between AARs using constant step-size parameters and (convergence-driven) decreasing step-size parameters in (5)

| | | Unicredit | Intesa | Tim |
|---|---|---|---|---|
| Sharpe | QL | 3.33 | 1.55 | 1.74 |
| | SARSA | 3.06 | 1.43 | 1.66 |
| | Greedy-GQ | 4.32 | 2.32 | 2.22 |
| Calmar | QL | 3.15 | 1.65 | 2.05 |
| | SARSA | 2.85 | 1.65 | 1.98 |
| | Greedy-GQ | 4.51 | 2.47 | 2.75 |

drawdown and of the effective Calmar ratio for the FTSs which achieved the best AAR, for each stock and for the two reward metrics. Moreover, for comparative purposes, we also show for each stock the AARs achieved by the simple investment strategy *Buy & Hold* (B&H). Note that in some cases FTSs which use the Calmar ratio show higher drawdown than FTSs using the Sharpe ratio. This suggests that in RL framework the classical financial measures of risk should be considered with care when used as reward metrics. Note also that for each stock the B&H AAR is negative.

Furthermore, we compared the results obtained using the setting with constant step-size parameters, with the ones obtained by imposing convergence-driven decreasing values. The results are shown in Table 2 in terms of the ratio between AARs in the former setting and in the latter. We always get best results with the constant choice of the step-size parameters, which confirms the non-stationarity based hypothesis of the distribution of rewards. We have reported the result only for three of the considered stocks, since for the remaining two ones the average equity obtained with decreasing step-size parameters was lower then the initial capital.

# References


1. Barto, A.G., Sutton, R.S.: Reinforcement Learning: An Introduction. The MIT Press, Boston (2018)
2. Bellman, R.E.: Dynamic Programming. Princeton University Press, Princeton (1957)
3. Bloomberg Finance L.P.: https://www.bloomberg.com/professional/product/market-data/
4. Corazza, M., Sangalli, A.: Q-learning and SARSA: a comparison between two intelligent stochastic control approaches for financial trading. Working Papers, Department of Economics, Ca' Foscari University of Venice, 15 (2015)
5. Geramifard, A., Dann, C., How, J.P.: Off-policy learning combined with automatic feature expansion for solving large MDPs. In: Proceedings of the 1st Multidisciplinary Conference on Reinforcement Learning and Decision Making, pp. 29–33. Princeton University Press, Princeton (2013)
6. Gosavi, A.: Simulation-Based Optimization. Parametric Optimization Techniques and Reinforcement Learning. Springer, Berlin (2015)
7. Lo, A.W.: Adaptive Markets. Financial Evolution at the Speed of Thought. Princeton University Press, Princeton (2017)
8. Maei, H.R., Szepesvári, C., Bhatnagar, S., Sutton, R.S.: Toward off-policy learning control with function approximation. In: International Conference on Machine Learning (ICML), pp. 719–726. Omnipress, Madison (2010)
9. Rummery, G.A., Niranjan, M.: On-line Q-Learning using connectionist systems. Technical Report CUED/F-INFENG/TR, 166, Engineering Department, Cambridge University (1994)
10. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**, 279–292 (1992)

# Author Queries

**Chapter 22**

| Query Refs. | Details Required | Author's response |
| --- | --- | --- |
| AQ1 | As keywords are mandatory for this chapter, please provide 3–6 keywords. | |

# MARKED PROOF

## Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

| *Instruction to printer* | *Textual mark* | *Marginal mark* |
|---|---|---|
| Leave unchanged | · · · under matter to remain | |
| Insert in text the matter indicated in the margin | | New matter followed by or |
| Delete | / through single character, rule or underline or through all characters to be deleted | or |
| Substitute character or substitute part of one or more word(s) | / through letter or through characters | new character / or new characters / |
| Change to italics | under matter to be changed | |
| Change to capitals | ≡ under matter to be changed | ≡ |
| Change to small capitals | = under matter to be changed | = |
| Change to bold type | under matter to be changed | |
| Change to bold italic | under matter to be changed | |
| Change to lower case | Encircle matter to be changed | |
| Change italic to upright type | (As above) | |
| Change bold to non-bold type | (As above) | |
| Insert 'superior' character | / through character or where required | or under character e.g. or |
| Insert 'inferior' character | (As above) | over character e.g. |
| Insert full stop | (As above) | |
| Insert comma | (As above) | |
| Insert single quotation marks | (As above) | or and/or or |
| Insert double quotation marks | (As above) | or and/or or |
| Insert hyphen | (As above) | |
| Start new paragraph | | |
| No new paragraph | | |
| Transpose | | |
| Close up | linking characters | |
| Insert or substitute space between characters or words | / through character or where required | |
| Reduce space between characters or words | between characters or words affected | |