Chapter 18
A Q-Learning Approach for Investment Decisions

Martín Varela, Omar Viera, and Franco Robledo

Abstract This work deals with the application of the Q-learning technique in order to make investment decisions. This implies to give investment recommendations about the convenience of investment on a particular asset. The reinforcement learning system, and particularly Q-learning, allows continuous learning based on decisions proposed by the system itself. This technique has several advantages, like the capability of decision-making independently of the learning stage, capacity of adaptation to the application domain, and a goal-oriented logic. These characteristics are very useful on financial problems.

Results of experiments made to evaluate the learning capacity of the method in the mentioned application domain are presented. Decision-making capacity on this domain is also evaluated.

As a result, a system based on Q-learning that learns from its own decisions in an investment context is obtained. The system presents some limitations when the space of states is big due to the lack of generalization of the Q-learning variant used.

Keywords Reinforcement learning • Q-learning • Portfolio selection • Artificial intelligence script • Machine learning • Finance • Investment decisions • Meta-heuristics • Technical analysis

18.1 Introduction

18.1.1 Context

The portfolio selection problem is a highly combinatorial problem with two opposite goals: maximize profit and minimize risk. There are many works that cover this problem with different approaches, being the modern portfolio theory proposed by Harry Markowitz in 1952 (Markowitz 1959), a milestone for further work until today.

M. Varela (✉) • O. Viera • F. Robledo
Facultad de Ingeniería, Universidad de la República, Julio Herrera y Reissig 565, 11300 Montevideo, Uruguay
e-mail: martinv@fing.edu.uy; viera@fing.edu.uy; frobledo@fing.edu.uy

A.A. Pinto et al. (eds.), *Trends in Mathematical Economics*,
DOI 10.1007/978-3-319-32543-9_18

From the very existence of the stock exchange, there exists a discussion about the predictability of the market, or more precisely, if the future price of an asset can be predicted. Efficient market hypothesis (Fama 1965, 1970) states that in an efficient market, all existing information is reflected in the price, so the current price is the best measure of its intrinsic value, and as consequence of this, it is impossible to predict its future behavior. Moreover, some argue that the market has enough inefficiencies to get a better profit based on speculation. The technical analysis is a widely used tool by who defend this position, since it provides mechanisms to deal with the historical series of price and volume of transactions, giving more digested information that is used to make market predictions. A strong argument of those who believe that the market behavior can be predicted is the existence of buy and sell behavioral patterns associated with psychological factors.

18.1.2 Motivation

Unlike many decision problems in which obtaining historical information is a difficult obstacle to overcome, in this problem the available information is abundant. It is not only available historical information, but it is possible to obtain near real-time information and for free. Moreover, this availability of information has been rising over time, which has meant that there are many more market participants, not always with a high level of knowledge about financial theories. Furthermore, this widespread growth tends to increase. This fact enforces the theory contrarian to the efficient market hypothesis, in the sense that a major number of participants with limited knowledge about financial theories generate possibilities of more inefficiencies, guided by a behavior more related with human psychology factors that with rationality.

While the application of several techniques and models to solve the portfolio selection problem has been studied, they have not been found in the context of this work, i.e., studies on the application of decision-making techniques that could adapt automatically to market changes. In this sense, reinforcement learning (Sutton and Barto 2000) results in a very interesting technique and its application to this domain seems to be novel.

18.1.3 Scope

This work pretends to advance in the treatment of the complex problem of investment decisions, providing a learning and decision-making tool based on historical and real-time information. Specifically, there is presented a system of recommendation on whether or not invest in a particular asset, being the result of that potential investment the feedback to the system, which allows it to learn and provide better recommendations in the future.

18.1.4 Organization

The work is divided into five sections. In the first one, the application domain is described, covering some important concepts like portfolio management and technical analysis. While portfolio management is mentioned for context reasons, technical analysis plays an important role in this work. In Sect. 18.2, reinforcement learning and particularly Q-learning is explained. In the third section, the aim of this work, the application of Q-learning to investment decisions, is justified and explained with detail. In the fourth, experimental results are discussed, and finally, conclusions and some lines of future work are described.

18.2 About the Application Domain

18.2.1 Portfolio Management Problem

In a market with m stocks, let $v_t = (v_t(1), \ldots, v_t(m))$ be the vector that contains the closing price of the m stocks at the day t. One way to become independent of individual prices of each stock is working with **relative prices** $x_t(j) = v_t(j)/v_{t-1}(j)$, that is, the relationship between the closing prices of two consecutive days. Thus, an investment of $\$d$ in the stock j made between day $t-1$ and day t will result in $\$dx_t(j)$ and is then denoted as $x_t = (x_t(1), \ldots, x_t(m))$ to the vector of relative prices for the m stocks at the day t. A **portfolio** b represents an allocation of weights to stocks, specified proportionally to the amount of money invested. Therefore b is expressed as $b = (b(1), \ldots, b(m))$, where $b(m) \geq 0$ and $\sum_j b(j) = 1$. The **daily return** of a portfolio b subject to the relative prices vector x is $bx = \sum_j b(j)x(j)$, and the **total return** $ret_X(b_1, \ldots, b_n)$ of a sequence of relative prices vectors $X = x_1, \ldots, x_n$ is $\prod_{t=1}^{n} b_t x_t$. It is called **portfolio selection algorithm** to any strategy to specify a sequence of portfolios (Borodin et al. 2004).

18.2.2 Modern Portfolio Theory

The **modern portfolio theory**, proposed by Harry Markowitz in 1952 (Markowitz 1959), proposes two conflicting objectives: maximize return and minimize risk. In this sense, an investment in a risky asset is only justified by a bigger expected return.

The model assumes that only asset expected return and volatility are relevant to investors. Volatility represents the risk, while the expected return is calculated as the average of historical returns.

The portfolio return is calculated as the weighted sum, according to the relative weight, of the returns of the assets comprising the portfolio. The portfolio volatility is a function of the correlation between assets comprising the portfolio. Expressed in mathematical equations,

$$\text{Expected return: } EV(x_{t+1}(P)) = \sum_i b_t(i)EV(x_{t+1}(i)) \tag{18.1}$$

$$\text{Portfolio variance: } \sigma_t^2(P) = \sum_i \sum_j b_t(i)b_t(j)\sigma_t(i)\sigma_t(j)\rho(i,j) \tag{18.2}$$

$$\text{Portfolio volatility: } \sigma_t(P) = \sqrt{\sigma_t^2(P)} \tag{18.3}$$

The mathematical model of variance minimization proposed by Markowitz is as follows:

$$\begin{cases} \text{min} \quad \sigma_t^2(P) = \sum_i \sum_j b_t(i)b_t(j)\sigma_t(i)\sigma_t(j)\rho(i,j) \\ \text{s.a.} \\ \qquad \sum_i b_t(i)EV(x_{t+1}(i)) \geq 0 \\ \qquad \sum_i b_t(i) \leq 0 \\ \qquad b_t(i) \geq 0 \;, \forall i \end{cases} \tag{18.4}$$

In this model, a quadratic problem must be solved. The objective is to minimize the portfolio variance while the expected return acts as a constraint.

18.2.3 Technical Analysis

Technical analysis is the study of market action, primarily through the use of charts, for the purpose of forecasting future price trends, Murphy (1999).

The market action includes the three main sources of information available for technicians: price, volume, and open interest (this is only used for futures and options).

Technical analysis is based on three premises:

1. Market action discounts everything.
2. Price moves in trends.
3. History tends to repeat itself.

The first premise indicates that any factor (financial, psychological, political, etc.) that could affect an asset price is already reflected on price, so the study of price action is enough to make right forecasts.

Based on the second premise, the objective is to identify trends early, so as to trading in the trend direction.

The study of market behavior is concerned with the study of human psychology. In a century of information about market behavior, behavioral patterns of buying and selling have been identified. Since these patterns worked well in the past, it is assumed that they will work well in the future. These patterns are based on the study of human psychology, which tends not to change. The third premise is based on this foundation.

There are, of course, some criticisms of the technical approach:

1. *Self-fulfilling prophecy*: This criticism suggests that given the growth in the use of these techniques in recent years, there are many traders using them, taking similar decisions massively, affecting the market and thus generating the expected movement. This statement seems somewhat simplistic in that it assumes that based in the same graph, the vast majority of traders will act the same way. Typically, the information resulting from technical analysis is not so clear, and every analyst makes a subjective interpretation. There are too many indicators and techniques as to assume that technical analysts behave in the same way.
2. *Can the past be used to predict the future?* Any known prediction method, whatever its application domain, is based on past data. There not exists another source of information to predict the future, that is, the knowledge about the past. This criticism is equally applicable to any prediction mechanism, including fundamental analysis. Regardless, what determines the price is the relation between supply and demand. Supply and demand respond to belief of investors about the future behavior of the asset. Investor beliefs rely on their knowledge, which is directly related to the past. Therefore, it seems reasonable to think that past experiences are a very good base for predicting future behavior.

18.3 Reinforcement Learning

18.3.1 Classification of Application Domain

Russell and Norvig (1995), Russell and Norvig (2003), and Russell and Norvig (2010) categorize application domains for decision-making systems based in the following properties:

- **Fully observable** or **partially observable**: If it is possible to know the complete state of the system at each point in time, then the domain is fully observable. If, instead, just some relevant information can be accessed, the domain is partially observable. If it is not possible to obtain any information about the domain at all, the domain is **unobservable**.
- **Deterministic** or **stochastic**: If the next state of the system is completely determined by the current state and the taken action, the domain is deterministic. Otherwise, it is stochastic.
- **Episodic** or **sequential**: An application domain is episodic if the experience can be divided into "episodes". Actions executed on an episode do not influence on the states of the following episodes. The last state of an episode is called **terminal state**. In sequential domains, the current decision could affect all future decisions.

- **Static** or **dynamic**: If the system state can change during the decision-making process, the domain is dynamic. Otherwise, it is a static domain.
- **Discrete** or **continuous**: If the union of the sets of state variables and possible actions result in a discrete set, then the domain is also discrete. If it is not the case, the domain is continuous.

The most difficult domains to work are those partially observable (or unobservable), stochastic, sequential, dynamic, and continuous.

18.3.2 Definition

The problem of reinforcement learning (Russell and Norvig 1995, 2003, 2010; Sutton and Barto 2000) is a simplistic abstraction of the interactive learning to achieve a goal. In this model, the apprentice and decision-maker is called agent, while all that which interact with the agent is called **environment**.

Reinforcement learning (Russell and Norvig 1995, 2003, 2010; Sutton and Barto 2000) tries to relate situations with actions in order to maximize a reward. Unlike supervised learning, where learning is based on samples provided by an external supervisor, here the agent is not told about which action must be taken in a determined situation, but is the agent itself who must discover which actions generate better rewards in each state, following a trial-and-error approach. This property and the fact that a reinforcement, positive or negative, is not associated with one only action but with a sequence of actions, are the most important factors to distinguish reinforcement learning from another learning techniques.

A reinforcement learning agent can start to work without previous knowledge, taking random decisions at first and learning from the reinforcements received as result of its actions. To make this possible, it is essential to maintain a balance between exploitation and exploration. In other words, as the system must learn from its own actions, sometimes it has to sacrifice an immediate positive reinforcement in order to explore the space of states with the hope of finding even better reinforcements.

A reinforcement learning system is composed by four main elements:

- **Policy**: Defines the agent behavior at a given time. It is a mapping between states and actions to be taken in those states. The policy may be a simple lookup table or it may involve a complex computational process.
- **Reward function**: The reward function is where the objective of a reinforcement learning problem is defined. This function maps the states or state-action pairs with a numerical value called reward. The intrinsic goal of a reinforcement learning agent is to maximize the total reward it receives in the long run. The reward function must be fixed, but the rewards received by it can be used to modify the policy.
- **Value function**: The value function estimates the goodness of a state in the long run, that is, the expected future accumulated reward starting from a given state.

State values estimation has a central role in reinforcement learning since the agent must try to reach those states with bigger value.

- **Model of the environment** (optional): The model of the environment tries to explain its behavior. Using this model the agent could predict the next state and the reward it will receive. The model of the environment is useful to incorporate planning into the agent decision-making process.

18.3.3 Agent-Environment Interaction

The agent and the environment interact continuously, the agent selecting actions and the environment responding to that actions and presenting new situations to the agent. The environment answer to the actions taken by the agent in form of rewards, numerical values that the agent will seek to maximize in the long run.

Interaction between the agent and the environment can be divided in a sequence of discrete steps. In every time step t, the agent receives a representation of the state of the environment, $s_t \in S$ (S is the set of possible states), and has to choose an action $a_t \in A(s_t)$ ($A(s_t)$ is the set of possible actions that can be taken in the state s_t). In the next time step $(t + 1)$, partially as a consequence of the action a_t, the agent receives a numerical reward, $r_{t+1} \in R$, and a representation of the new state of the environment s_{t+1}. A diagram of this interaction is showed in Fig. 18.1.

The rewards give an indication about how good or bad are the actions taken earlier. The agent implicit objective is to maximize accumulated rewards in the long run. The mechanism of rewards indicates to the agent the goals it must pursue, but not how. This is a distinguishing factor of reinforcement learning.

Formally, the goal of a reinforcement learning agent is to maximize the **expected return**, where the return R_t is defined as a function of the sequence of returns. The simplest example is calculating the return as a sum of rewards: $R_t = r_{t+1} + r_{t+2} + \ldots + r_T$, where T is a final time step. This calculation has sense when the problem can be divided into independent subsequences called episodes. When the problem cannot be divided into episodes and the interaction between the agent and

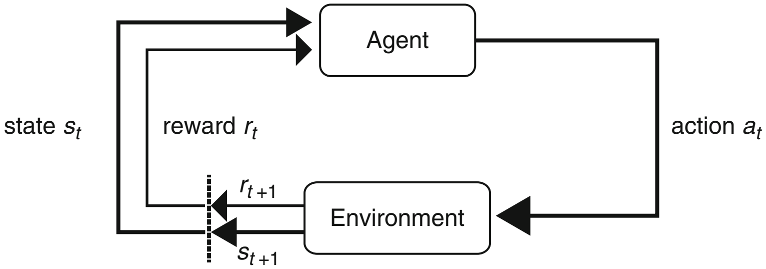

Fig. 18.1 Interaction between the agent and the environment in a reinforcement learning problem (Sutton and Barto 2000)

the environment continues to infinity, this equation is not valid because there is not a final time step T. A solution to this issue is to add the concept of **discount**. Thus, the agent will try to maximize the expected **discounted return**. The discounted return is expressed by the following equation: $R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, where $\gamma \in [0, 1]$ is the **discount factor**, a parameter that adjusts the value of a future reward depending on the number of time steps to get the reward.

18.3.4 Value Functions

Reinforcement learning algorithms are, in the most, based on estimating **value functions**. Value functions depend on state (or state-action pair) and give an estimation about the goodness of reaching a particular state (or the goodness of taking a particular action in a particular state). The goodness is based on the expected future rewards, which depend on the actions taken by the agent. Therefore, the value functions are defined regarding particular policies.

A policy π maps for every pair of state $s \in S$ and action $a \in A(s)$, the likelihood of taking the action a in the state s. The value of a state s under a policy π, $V^{\pi}(s)$, is the expected return starting in the state s and following from there the policy π.

Formally, $V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\}$, where $E_{\pi}\{\}$ denotes the expected value following the policy π. V^{π} is called **state-value function for policy** π.

Similarly, the value of selecting a particular action a in a state s under a policy π, $Q^{\pi}(s, a)$, is defined as the expected return starting in the state s, selecting the action a, and then following the policy π: $Q^{\pi}(s, a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\}$. Q^{π} is called **action-value function for policy** π.

The value functions V^{π} and Q^{π} can be estimated from experience.

A fundamental property of value functions is that satisfy the **Bellman equation for** V^{π}:

$$\begin{aligned}
V^{\pi}(s) &= E_{\pi}\{R_t \mid s_t = s\} \\
&= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\} \\
&= E_{\pi}\left\{r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s\right\} \\
&= \sum_{a \in A(s)} \pi(s, a) \sum_{s' \in S} P^a_{ss'} \left[R^a_{ss'} + \gamma E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s'\right\}\right] \\
&= \sum_{a \in A(s)} \pi(s, a) \sum_{s' \in S} P^a_{ss'} [R^a_{ss'} + \gamma V^{\pi}(s')] \qquad (18.5)
\end{aligned}$$

where $P^a_{ss'} = P\{s_{t+1} = s' \mid s_t = s, a_t = a\}$ is the likelihood of reaching the state s' from the application of the action a in the state s. Similarly, $R^a_{ss'} = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\}$ is the expected value of the reward to obtain for reaching the state s' after applying the action a in the state s.

Bellman equation expresses a relation between the value of a state and the value of the successor states, weighing every possible future reward with the likelihood of obtaining it, considering the discount factor.

18.3.5 Temporal Difference Learning

Temporal difference (TD) (Fama 1965) is a combination of Monte Carlo and dynamic programming ideas.

TD and Monte Carlo have in common the use of experience to adjust its estimation of state values V. Monte Carlo methods wait to know the reward following the visit to a state, to update the value $V(s_t)$ using this return as an objective value of $V(s_t)$. An example of a Monte Carlo method is $V(s_t) \leftarrow V(s_t) + \alpha[R_t - V(s_t)]$, where R_t is the return obtained after time t and α is a learning factor that weighs the influence of a particular learning instance over the estimated state value $V(s_t)$. This method is called *constant-α MC*.

Unlike Monte Carlo methods, which must wait until the end of an episode to update $V(s_t)$ because the value of R_t is unknown till then, TD methods just have to wait until the next time step. The TD method known as **TD(0)** updates the value $V(s_t)$ according to the following equation:

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \tag{18.6}$$

As can be observed, TD uses $r_{t+1} + \gamma V(s_{t+1})$ as an alternative of the Monte Carlo method objective R_t, based on the following deductive logic:

$$\begin{aligned} V^\pi(s) &= E_\pi\{R_t \mid s_t = s\} \\ &= E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\} \\ &= E_\pi\left\{r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s\right\} \\ &= E_\pi\{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s\} \end{aligned} \tag{18.7}$$

As $V^\pi(s_{t+1})$ is not known at time t, TD uses its current estimation $V_t(s_{t+1})$ instead of it.

So, TD methods combine the sample-based learning of Monte Carlo methods with the iterative logic of dynamic programming, which has the advantage versus Monte Carlo methods of not requiring waiting until a terminal state to learn, and versus dynamic programming of not requiring a model of the environment.

18.3.6 Q-Learning

The Q-learning algorithm (Watkins 1989) is a control algorithm based on TD learning. Instead of estimating every state value, this algorithm estimates every state-action pair value. State-action pair values are updated according to the following equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{18.8}$$

The Q-learning algorithm pseudocode is shown below:

Algorithm 1 Q-learning algorithm pseudocode

```
Initialize $Q(s, a)$ arbitrarily
for each episode do
    Initialize $s$
    repeat
        Take action $a \in A(s)$ based on $Q(a, s)$ values (e.g., $\epsilon$-greedy)
        Observe $r$ and $s'$
        $Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma \max_{a' \in A(s')} Q(s', a') - Q(s, a) \right]$
        $s \leftarrow s'$
    until $s$ is terminal
end for
```

This algorithm converges to optimal policy and optimal state-action pair values insofar as every state-action pair is evaluated infinite times and that the policy converges to the greedy policy (eliminating exploration).

18.4 Q-Learning for Investment Decisions

18.4.1 Introduction

The portfolio selection problem can be divided into several subproblems. Some are:

1. Determining the expected return of a particular asset
2. Determining the risk associated with a particular asset
3. Determining the best portfolio composition in order to maximize the expected return and minimize the risk

Each subproblem mentioned above can imply a lot of work given the complexity of them. The scope of this work is a mix of subproblems 1 and 2, looking to develop a system capable of deciding if it is convenient or not to invest in a particular asset. While this approach differs from subproblems 1 and 2 in that the system does not return an expected return neither a risk associated with the investment, these elements are implicitly considered by the system in order to determine the investment convenience.

In addition to building a system capable to make good recommendations, the learning capability of the system has a big emphasis in this work. It is pretended to build an agent capable of learning from the results of its own recommendations, improving in this way its future decisions.

In this sense, reinforcement learning, and particularly the Q-learning variant, offers an appropriated theoretical framework for the treatment of this problem. Here are some properties of this learning and decision-making mechanism that make it appropriate to be applied in this context:

- It is suitable for stochastic problems
- It is suitable for non-episodic problems
- It is suitable for dynamic problems
- It is suitable for nonstationary problems
- It is possible to take decisions without prior knowledge
- Adaptation capability (continuous learning without between training and execution)

The application of Q-learning for investment decisions implies decisions at three levels: the generation of a model of the environment, determination of a learning mechanism, and determination of a decision-making mechanism.

18.4.2 Model of the Environment

Defining the model of the environment implies, basically, selecting the variables that compose the state of the environment and the set of actions that the agent can take in each state. The system of reward determination can also be considered part of the model of the environment.

In this model it is considered that the agent acts daily, so the time is discrete and every time step represents 1 day.

As the agent will make investment recommendations on a single asset, the state variables are the result of different technical analyses applied to this asset. The available data to define the state variables are those that summarize the activity of a day: Open, Close, Low, High, and Volume.

A lot of technical analysis and even combination of these were analyzed, but finally, after studying the individual behavior of each analysis and the correlation between them, the following set of 18 variables were selected to compose the state of the environment:

- Closing price and moving averages of closing price
  - Closing price is higher or lower than previous closing price.

  - The 20-day moving average goes up or down from the previous day.
  - The 50-day moving average goes up or down from the previous day.
  - The 200-day moving average goes up or down from the previous day.

- Relative strength index (RSI)
  - RSI value is in the interval $[0, 30)$, $[30, 70]$, or $(70, 100]$.
  - RSI value goes up or down from the previous day.
- Moving average convergence/divergence (MACD)
  - A range is determined from the minimum and maximum MACD values in the last year. MACD value is positive and greater than 80 % of the defined range, is positive and less than 80 % of the defined range, is negative and greater than 20 % of the defined range, or is negative and less than 20 % of the defined range. That is, from the range, MACD value is in the interval $[-\infty, \text{range} \times 0.2)$, $[\text{range} \times 0.2, 0)$, $[0, \text{range} \times 0.8]$, or $[\text{range} \times 0.8, \infty)$.
  - MACD value goes up or down from the previous day.
  - MACD histogram value goes up or down from the previous day.
  - MACD signal line value goes up or down from the previous day.
- Stochastic oscillator (K%D)
  - Slow %K value is in the interval $[0, 20)$, $[20, 50)$, $[50, 80]$, or $(80, 100]$.
  - Slow %K value goes up or down from the previous day.
  - Slow %D value goes up or down from the previous day.
- Slopes of closing price moving averages (slopes are calculated as the difference between current value and 10 days before value)
  - The 10-day moving average of the slope of the 20-day moving average of closing price goes up or down from the previous day.
  - The 25-day moving average of the slope of the 50-day moving average of closing price goes up or down from the previous day.
  - The 100-day moving average of the slope of the 200-day moving average of closing price goes up or down from the previous day.
- Open, Close, Low, High
  - Close is higher or lower than Open.
- Volume
  - Volume of transactions goes up or down from the previous day.

An additional variable is added to indicate if the agent has money invested on the asset or it has not.

The actions that the agent can take are three:

- Buying: Implies investing all available money on the asset
- Selling: Implies selling the totality of the asset shares
- Do nothing: The agent maintains the number of shares of the previous step

At each time step, the agent can choose an action depending on the investment state. If the agent has the money invested on the asset, it can sell the shares it owns or do nothing. If the agent has not invested the money, it can buy shares or do nothing. So, at each time step, the agent can choose one of two possible actions: $sell, donothing$ or $buy, donothing$.

The reward system selected is based on the investment return. It is not considered the money that the agent win or lose, but the return of the investment measured as $\text{ret}_t \leftarrow (\text{Close}_t/\text{Close}_{t-1}) - 1$. The agent is rewarded based on this equation only when it owns shares of the asset.

18.4.3 Learning Mechanism

The learning mechanism emerges from the subject of this work: Q-learning. However, there are alternatives for Q-learning application. The classical variant allows to work only with a discrete space of states and actions, while the problem being solved is continuous in nature. While there is research on the adaptation of Q-learning for the treatment of continuous problems (i.e., the use of neural networks for estimating Q-values), it was understood in this work that, for a first approach to this problem using Q-learning, it was more convenient to apply the classical variant of this learning technique, which is based on the storage of the Q-value of each state-action pair, considering that it has a stronger theoretical framework, taking into account its limitations for continuity and state generalization treatment. So, the learning of each state-action pair is based on the following equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{18.9}$$

where s_t and a_t are the state and the chosen action at time t, respectively, r_{t+1} is the reward obtained at time $t + 1$, and s_{t+1} and $A(s_{t+1})$ are the state and the set of possible actions to be taken at time $t + 1$, respectively, α the learning factor and γ the discount factor.

18.4.4 Decision-Making Mechanism

There are several alternatives for decision-making, being the equilibrium between exploitation and exploration a fundamental factor. It was selected in this work as an adaptation of an algorithm based on Ant Colony Systems (Dorigo and Gambardella 1997), which are widely used to solve transportation problems and their learning mechanism can be classified as reinforcement learning.

The adaptation made comes to solve the issue of negative reinforcements, which are not considered on its original formulation. The adaptation consists of dividing the set of possible actions in two sets, one containing the actions with positive Q-values and the other containing the actions with negative Q-values. Then, with a probability p (a system parameter), a positive Q-value action is selected, and with probability $1 - p$, a negative Q-value action is selected. On each case, positive or negative, the action is taken with likelihood proportional to its Q-value. Dorigo and Gambardella (1997) also propose on their work a heuristic factor designed for transportation problem. This factor is not considered in this work.

The pseudocode of the decision-making algorithm proposed on this work is shown below:

Algorithm 2 Decision-making algorithm pseudocode proposed in this work

```
$r =$ A random number between 0 and 1
**if** $r \leq 1 - \epsilon$ **then**
    Select $a \in A(s)$ such that $Q(s,a) = \max_{a' \in A(s)} Q(s,a')$
**else**
    Select $a \in A(s)$ as follows
    $A^{+}(s) = \{a \in A(s) \mid Q(s,a) \geq 0\}$ (set of actions with positive Q-value)
    $A^{-}(s) = \{a \in A(s) \mid Q(s,a) < 0\}$ (set of actions with negative Q-value)
    **if** $A^{+}(s)$ and $A^{-}(s)$ are not empty **then**
        With probability $p$ select $a \in A^{+}(s)$ with probability $Q(s,a) / \sum_{a' \in A^{+}(s)} Q(s,a')$
        With probability $1 - p$ select $a \in A^{-}(s)$ with probability $\frac{1}{-Q(s,a)} / \sum_{a' \in A^{-}(s)} \frac{1}{-Q(s,a')}$
    **else**
        **if** $A^{-}(s)$ is empty **then**
            Select $a \in A^{+}(s)$ with probability $Q(s,a) / \sum_{a' \in A^{+}(s)} Q(s,a')$
        **else**
            Select $a \in A^{-}(s)$ with probability $\frac{1}{-Q(s,a)} / \sum_{a' \in A^{-}(s)} \frac{1}{-Q(s,a')}$
        **end if**
    **end if**
**end if**
```

As can be seen, parameters ϵ and p determine the exploration level of the agent.

18.5 Experimental Results

The tests performed on the developed system had two main objectives:

1. Evaluating the learning capacity
2. Evaluating the system ability to take good decisions

For the first objective, the analysis is centered on the agent behavior while it is acquiring more experience. To make this analysis, a graph of the average of rewards obtained as a function of experience is used.

For the second objective, it is necessary to make a comparison against some measure that belongs to the application domain. In this sense, a widely used measure is the result of applying the B&H strategy. The graph used to make this analysis shows the economic profit made by the Q-learning agent (Q-agent) versus the economic profit following the B&H strategy.

Some assumptions were made in order to simplify evaluation:

- Actions taken by the agent are always executed.
- Transaction costs were not considered.
- The asset price does not change while the action of buying and selling is being executed.

Preliminary tests were performed in order to determine the values of the Q-agent parameters. The results showed here are based on the following parameter values: $\alpha = 0.1$, $\gamma = 0.9$, $\epsilon = 0.5$, and $p = 0.9$.

The historical data used correspond to the Dow Jones Industrial Index between October 2, 1930 and December 12, 2010. The data was obtained from the Yahoo Finance Database (2011).

The tests were performed on a Toshiba Satellite L305-SP6924R Laptop, with an Intel Pentium Dual CPU T3400 2.16 GHz processor and 4 GB RAM, using Windows Vista 64 bits as operative system.

18.5.1 First Evaluation: Simultaneous Execution and Learning

In the first evaluation of the system, the agent began to take decisions from the first data of the time series (October 2, 1930), learning dynamically from the results of its own decisions. Therefore, the complete set of historical data was used in this evaluation.

Figure 18.2 shows the average of rewards obtained as a function of the experience.

As can be observed, the agent begins taking actions with negative reward, and as it gains experience, the rewards increase. A very important observation is that the average of rewards is positive once it stabilizes, so the agent achieves profit.

Figure 18.3 compares the accumulated return obtained by the Q-agent versus the return obtained by the B&H strategy.

A first hypothesis about the difference between both mechanisms could be that the Q-agent begins without knowledge, taking bad initial decisions and affecting in this way the accumulated return. But nevertheless, if the graph is cautiously observed, it can be seen that the accumulated returns do not show big differences until some point after the middle of the considered period. At this point the average of rewards received by the agent was already stable.

Analyzing more deeply the agent Q behavior, it was detected that the agent did not repeat each state many times, affecting clearly its learning capacity.

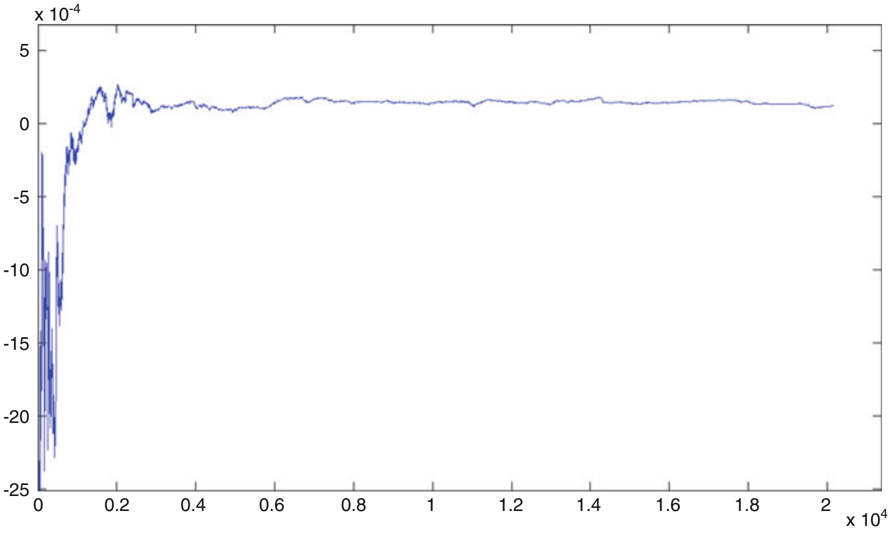

Fig. 18.2 Average of rewards obtained by the agent as a function of the experience

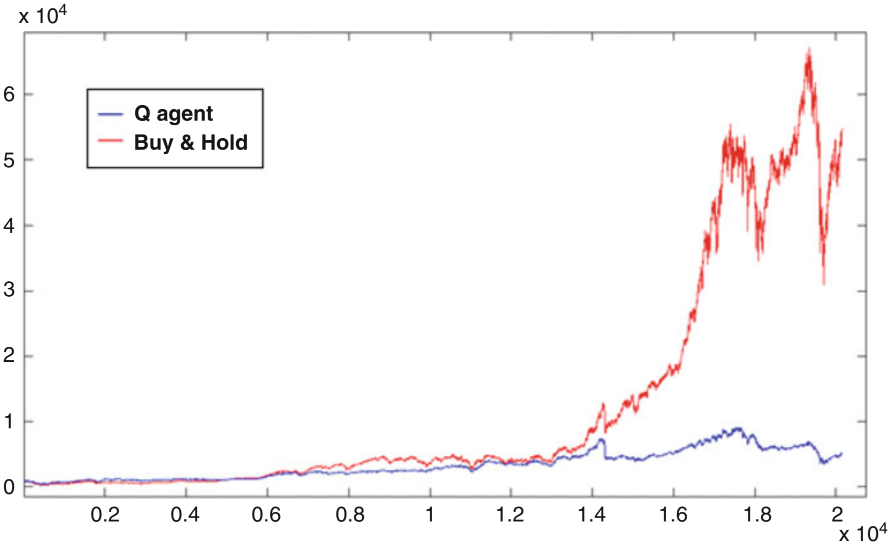

Fig. 18.3 Return obtained by the Q-agent against B&H

18.5.2 Second Evaluation: Training and Execution

As a way to solve the issue of the very little repeat level of states achieved by the agent, the historical data series was divided in two subsets: training data (10/02/1930–12/31/2004) and validation data (01/03/2005–12/31/2010).

This evaluation implies using the training set several times, with the goal of achieving a greater level of state repeat, and thus being able to learn more. This procedure introduces a risk of overtraining. That is why an independent set of data (validation data) is used to evaluate the agent behavior.

The training was therefore divided in episodes. In each episode the agent goes through the training set taking decisions and learning. The agent behavior was analyzed as a function of the number of training episodes.

Figure 18.4 shows the average of rewards received by the agents as a function of experience, for different numbers of training episodes.

As can be observed, as the number of training episodes increases, also the average of rewards increases. This fact shows that the agent is learning.

Figure 18.5 shows the accumulated returns obtained by the agent at different numbers of training episodes. These returns are compared against the B&H strategy.

As can be observed, the accumulated return obtained by the Q-agent fails to overcome the accumulated return obtained by B&H, but it does as the number of episodes increases. The behavior of the accumulated return as a function of the number of episodes is another indicator that the agent achieves learning.

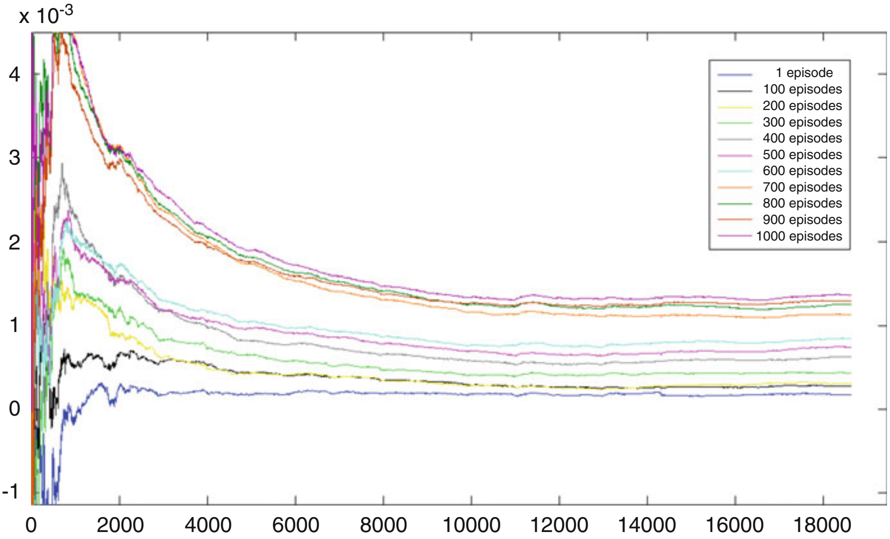

Fig. 18.4 Average of rewards as a function of experience, according to the number of training episodes

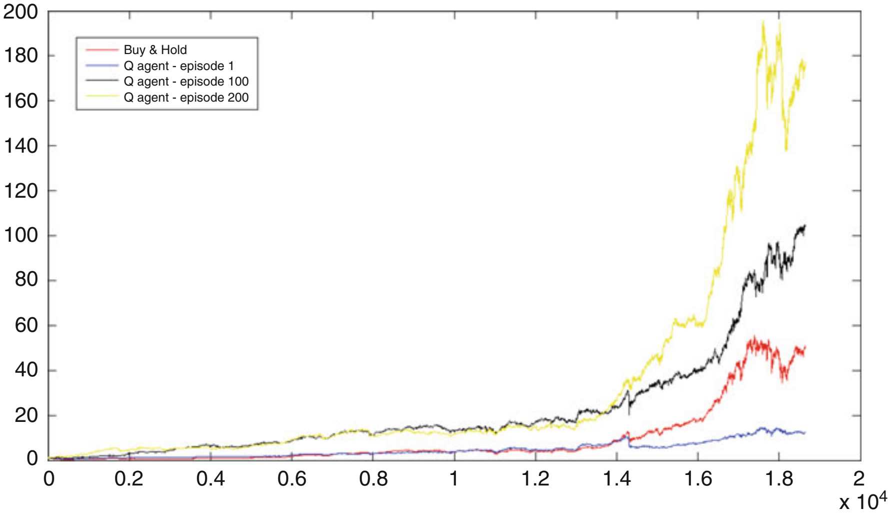

Fig. 18.5 Accumulated returns obtained by the Q-agent (at episodes 1, 100, and 200) compared against B&H

After verifying that the agent learns to make better decisions as it trains many times using the same set of data, the next step is to evaluate the system using the validation data.

Figure 18.6 shows the average of rewards received by the agents as a function of experience, for different numbers of training episodes using the validation data.

Figure 18.7 shows the accumulated returns obtained by the agent using the validation data, according to the number of training episodes and their comparison with B&H strategy.

From the observation of the last two graphs, it can be deduced that the behavior achieved with the training data is not achieved over the validation data. Unlike what happens with the training data, here the average of returns does not necessarily improve as the number of training episodes increases. The same behavior can be observed in relation with accumulated return.

A deeper analysis about the agent behavior showed that the most states reached by the agent during the evaluation phase were unknown to the agent (were never reached in the training phase), so the agent is making decisions without knowledge. This explains the excellent behavior achieved over the training data and the average performance obtained over the validation data.

The explanation about the unknown states is on the lack of generalization of the Q-learning algorithm used.

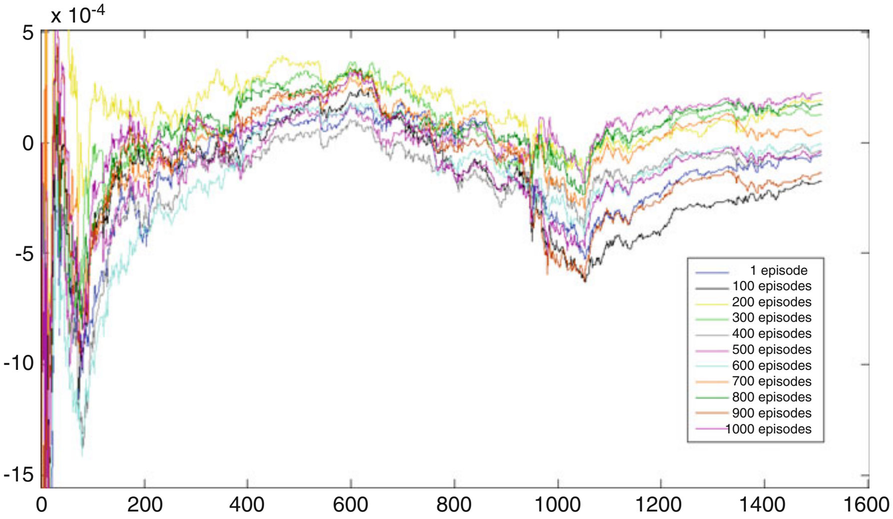

Fig. 18.6 Average of rewards as a function of experience, according to the number of training episodes, using validation data

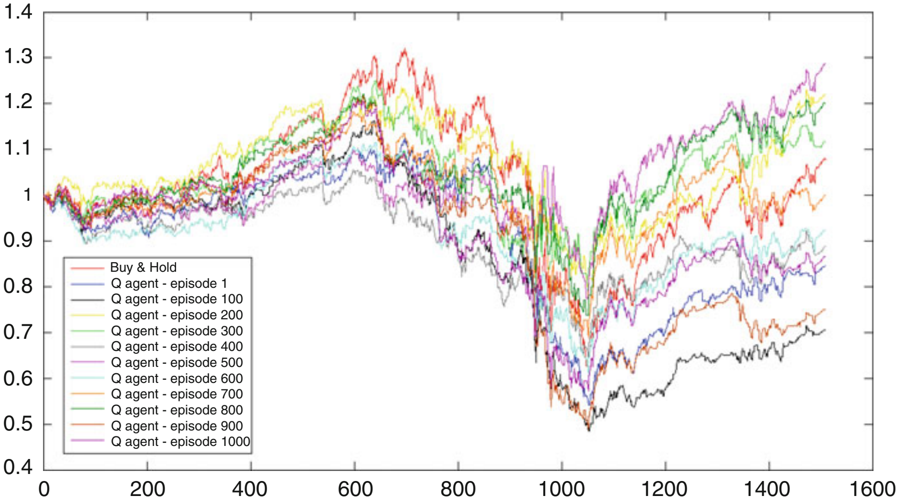

Fig. 18.7 Accumulated returns obtained by the Q-agent using validation data compared against B&H

18.6 Conclusions and Further Research

It was built as a learning and decision-making system based on Q-learning. This system was applied in an investment recommendation context using a discrete model. It was empirically shown that the agent achieves learning when it is applied with models that contain enough information of the environment.

However, it was not possible to generalize the learning achieved by the agent to future situations. According to the authors of this work, this may be due to several reasons:

1. It was not possible to generate a discrete model of the environment generic enough (composed for a little number of variables) to represent the information needed to make a forecast. In the experiments made, many models were tested. The bigger ones (many variables) did not allow generalization, while the small ones (a few variables) were not representative of the domain. The existence of a discrete model that allows the system to learn and generalize between states is a possibility, but it could not be generated in the context of this work.
2. The Q-learning algorithm used in this work is based on discrete models, storing the Q-values associated with every state-action pair in a lookup table. This implementation does not allow generalizing between similar states. This is, probably, the biggest limitation of this learning mechanism and, according to the authors of this work, the main reason for the difference of the agent behavior in training and validation phases. There are studies on the incorporation of generalization to the reinforcement learning mechanism (Bertsekas and Yu 2012; Maei et al. 2010; Precup et al. 2000; Rafols et al. 2005; Sutton 1999a,b; Sutton et al. 2000, 2009a,b; Van Hasselt 2012; Xu et al. 2014), which were not part of the scope of this work, but clearly indicate a line of future work. It was understood that for the first approach on the application of Q-learning to the investment decision problem, it was more convenient to use the tabular version of this mechanism, due to the bigger level of research about it, even knowing its limitations.
3. It is possible that the future behavior of asset price could not be predicted based on historical information. This is one of the theories about market prediction [efficient market hypothesis Fama (1965, 1970)].

From this work some lines of future work are proposed:

1. **Model of the environment**: The search for a model that represents enough information to determine the future market behavior, based on discrete or continuous variables, can be considered a line of further research. The sources of information are many, like technical analysis, fundamental analysis, expert opinion, financial news, etc.
2. **Q-learning generalization**: As was already mentioned, the tabular Q-learning algorithm does not allow generalizing between similar states. This constraint could be eliminated if another mechanism of Q-values estimation is considered. One possible solution is using function approximation techniques. In this way, instead of using Q-values stored in a table, there are estimated values based on a function built by approximation from a set of samples. These samples could be the own experience of agent. An interesting example of function approximation technique that could be applied to add generalization to Q-learning is neural networks. Some examples in this line of research can be found in Bertsekas and Yu (2012), Maei et al. (2010), Precup et al. (2000), Rafols et al. (2005),

Sutton (1999a), Sutton (1999b), Sutton et al. (2000), Sutton et al. (2009a), Sutton et al. (2009b), Van Hasselt (2012), and Xu et al. (2014). This line of future work includes the treatment of higher-dimensional models of the environment.

3. **Portfolio selection strategies**: This work was limited to the study of the application of continuous learning and decision-making mechanisms for investment recommendations on a single asset. How to insert a system like this into a portfolio selection strategy, considering recommendations on many assets as well as risk measures, can be also considered a particular line of further research.
4. **Application of reinforcement learning to other financial problems**: Reinforcement learning, and particularly Q-learning, offers a very interesting theoretical framework for the treatment of decision problems oriented to short-term results. Financial problems, in general, have this characteristic. But nevertheless, it was not detected in the context of this work, the existence of an important number of works about the application of this mechanism in this field.
5. **Techniques to avoid overfitting**: When the same set of training data is used to train the system, like it was made in the present work, overfitting is a logical consequence. The use of techniques to avoid overfitting can give to this work a more complete coverage. Some ideas can be found in Whiteson et al. (2011).

References

Bertsekas, D.P., Yu, H.: Q-Learning and enhanced policy iteration in discounted dynamic programming. Math. Oper. Res. **37**(1), 66–94 (2012)

Borodin, A., El-Yaniv, R., Gogan, V.: Can we learn to beat the best stock. J. Artif. Intell. Res. **21**, 579–94 (2004)

Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evolutionary Computation **1**(1), 53–66 (1997)

Fama, E.F.: The behavior of stock-market prices. J. Bus. **38**(1), 34–105 (1965)

Fama, E.F.: Efficient capital markets: a review of theory and empirical work. J. Finance **25**(2), 383–417 (1970)

Maei, H.R., Szepesvari, C., Bhatnagar, S., Precup, D., Silver, D., Sutton, R.S.: Convergent temporal-difference learning with arbitrary smooth function approximation. In: Advances in Neural Information Processing Systems. 23rd Annual Conference on Neural Information Processing Systems, Vancouver, 7–10 December 2009, pp. 1204–1212. La Jolla (2010)

Markowitz, H.M.: Portfolio Selection. Yale University Press, New Haven (1959)

Murphy, J.J.: Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance, New York (1999)

Precup, D., Sutton, R.S., Dasgupta, S.: Off-policy Temporal-difference Learning with Function Approximation. In: Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001), June 2001, pp. 417–424. Morgan Kaufmann, San Francisco (2000)

Rafols, E.J., Ring, M.B., Sutton, R.S., Tanner, B.: Using predictive representations to improve generalization in reinforcement learning. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, July 2005, pp. 835–840. Professional Book Center, Denver (2005)

Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall/Pearson Education, Upper Saddle River (1995)

Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall/Pearson Education, Upper Saddle River (2003)

Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson, Boston (2010)

Sutton, R.S.: Open theoretical questions in reinforcement learning. In: Computational Learning Theory. Lecture Notes in Computer Science, vol. 1572, pp. 637–638. Springer, Berlin (1999)

Sutton, R.S.: Reinforcement learning: past, present and future. In: Simulated Evolution and Learning. Lecture Notes in Computer Science, vol. 1585, pp. 195–197. Springer, Berlin (1999)

Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT, Cambridge (2000)

Sutton, R.S., McAllester, D. Singh, S., Mansour, Y.: Policy Gradient Methods for Reinforcement Learning with Function Approximation. In: Advances in Neural Information Processing Systems, 1999 Conference, vol. 12, pp. 1057–1063. MIT, Cambridge (2000)

Sutton, R.S., Maei, H.R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., Wiewiora, E.: Fast gradient-descent methods for temporal-difference learning with linear function approximation. In: Proceedings, Twenty-sixth International Conference on Machine Learning, pp. 993–1000. Omnipress, Madison (2009a)

Sutton, R.S., Szepesvari, C., Maei, H.R.: A convergent o(n) algorithm for off-policy temporal difference learning with linear function approximation. In: Advances in Neural Information Processing Systems. 22nd Annual Conference on Neural Information Processing Systems, Vancouver, 8–10 December 2008, vol. 21 pp. 1609–1616, Curran, Red Hook (2009b)

Van Hasselt, H.: Reinforcement learning in continuous state and action spaces. In: Reinforcement Learning, pp. 207–251. Springer, Berlin (2012)

Watkins, C.J.C.H.: Learning from Delayed Rewards. University of Cambridge, Cambridge (1989)

Whiteson, S., Tanner, B., Taylor, M., Stone, P.: Protecting against evaluation overfitting in empirical reinforcement learning. In: Symposium on Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), pp. 120–127. IEEE, Paris (2011)

Xu, X., Zuo, L., Huang, Z.: Reinforcement learning algorithms with function approximation: recent advances and applications. Inform. Sci. **261**, 1–31 (2014)

Yahoo! Finance - Business Finance, Stock Market, Quotes, News. http://finance.yahoo.com/. Accessed 19 December 2011