International Journal of Statistics and Probability; Vol. 10, No. 2; March 2021
ISSN 1927-7032 E-ISSN 1927-7040
Published by Canadian Center of Science and Education

A Q-learning Approach to a Consumption-Investment Problem

Ruy López-Ríos[1] & Hugo Cruz-Suárez[1]

[1] Facultad de Ciencias Físico Matemáticas, Benemérita Universidad Autónoma de Puebla, Puebla, México

Correspondence: Ruy López-Ríos, Facultad de Ciencias Físico Matemáticas, Benemérita Universidad Autónoma de Puebla, Puebla de Zaragoza, Puebla, Mexico. E-mail: ruy.lopez@alumno.buap.mx

Received: December 29, 2020 Accepted: February 20, 2021 Online Published: February 23, 2021

doi:10.5539/ijsp.v10n2p110 URL: https://doi.org/10.5539/ijsp.v10n2p110

Abstract

The paper deals with a discrete-time consumption investment problem with an infinite horizon. This problem is formulated as a Markov decision process with an expected total discounted utility as an objective function. This paper aims to presents a procedure to approximate the solution via machine learning, specifically, a Q-learning technique. The numerical results of the problem are provided.

Keywords: consumption-investment problem, dynamic programming, Markov decision processes, Q-learning

1. Introduction

The problem of consumption-investment initially arose in work on the portfolio problem by (Samuelson, 1969). Samuelson considers an investor who consistently wants to increase wealth at each point in time and, wishes to allocate this wealth between investment and consumption. The objective is then to maximize the investor's utility consumption, where the investor chooses on the best consumption-investment strategy for allocating the total investment among various assets. See (Mendelssohn & Sobel, 1980; White, 1993; Cruz-Suárez, Montes-de-Oca, & Zacarías, 2011 and Vitoriano & Parlier, 2017).

The main objective of this study is to propose a procedure based on machine learning techniques to approximate the investment strategy in a consumption-investment problem. Machine learning is the study of computer algorithms that automatically improve through experience. Machine learning algorithms build a mathematical model based on sample data, known as training data, to make predictions or decisions without being explicitly programmed to do so. The Q-learning technique belongs to this class of algorithms, and it is a reinforcement learning algorithm that was introduced by Watkins in 1989 (Watkins, 1989). Q-learning is an asynchronous dynamic programming method that allows the controller learn to act optimally in Markovian domains. (Watkins & Dayan, 1992) prove that Q-learning converges to the optimum action values with unit probability. Therefore, Q-learning is an adequate technique to implement in the solution of the consumption-investment problem. In closely related studies, (Weissensteiner, 2009) investigated the optimal consumption problem with a finite planing horizon and the simulation of scenarios were obtained via a geometric Brownian motion.

Now, this work proposes using Q-learning combined with classic value iteration algorithms for Markov decision processes in discrete time with the Robbins-Monro stochastic approximation method (Robbins & Monro, 1951), defining and optimizing an auxiliary function, and, discretizing the elements of interest, achieving an approximation of the optimal policy. To this end, a logarithmic function is assumed for the utility of consumption, see (Liang & Ma, 2019), and the states of the system satisfy a discrete dynamic system (see (5), below). Moreover, we consider state space and a continuous action space, and dedicate a section to the process of discretizing such spaces so they can be adapted to Q-learning. The conditions that guarantee the existence of a solution are covered, for both the original problem and the convergence of the Q-learning approach. In is presented a consumption investment problem with finite horizon and a numerical solution via Q-learning approach is provided.

The paper is organized as follows. Section 2 states the problem, considering the source of the law of capital transition through the Langevin equation. Section 3 illustrates the process of discretization of state and action spaces. Section 4 presents the Q-learning method. Finally, in Section 5, the theory is implemented in a numerical example, see (Cruz-Suárez, Montes-de-Oca, & Zacarías, 2011).

2. Problem Statement

Let $\{W_t, t \geq 0\}$ be a Brownian motion with respect to a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$, where $\mathcal{F}_t = \sigma\{W_s : 0 \leq s \leq t\}$. Consider $\{Y_t, t \geq 0\}$ a stochastic process defined on $\{\mathcal{F}_t : t \geq 0\}$ that satisfies the following stochastic equation:

$$Y_t = Y_0 - \mu \int_0^t Y_s ds + \sigma \int_0^t dW_s, \quad t \in [0, T], \tag{1}$$

where $\mu, \sigma, T \in \mathbb{R}^+$, with initial condition $Y_0 = y \in \mathbb{R}$. Expression (1) is known as the *Langevin equation*. $\{Y_t, t \geq 0\}$ is the *Ornstein-Uhlenbeck* process with state space $\mathbb{R}$, see (Mikosch, 1998). Equation (1) can be rewritten as

$$dY_t = -\mu Y_t \, dt + \sigma \, dW_t, \tag{2}$$

where μ and σ are parameters that are commonly called the drift and diffusion coefficients, respectively, of process $\{Y_t\}$, for details see p. 136 in (Mikosch, 1998).

For approximation purposes, we require a partition of the interval $[0, T]$. Discretizing (2) on the interval $[0, T]$ using the Euler-Maruyama approximation method, we establish the following recursive formula, see (Higham, 2001):

$$Y_{t+1} = Y_t - \mu \, Y_t \Delta t \, + \, \sigma \, \Delta W_{t+1} \tag{3}$$

where $\Delta W_{t+1} := W_{t+1} - W_t$. For simplicity, we use the step interval $\Delta t = 1$, so that ΔW_{t+1} has a normal distribution with mean zero and variance one ($\Delta W_{t+1} \sim Nor(0, 1)$).

Taking the exponential of (3) and setting $x_t := e^{Y_t}$ then yields

$$x_{t+1} = x_t^{(1-\mu)} \, e^{\sigma \, \Delta W_{t+1}}.$$

Finally, with $\delta := 1 - \mu < 1$ and $\xi_t := exp(\sigma \, \Delta W_{t+1})$, we obtain

$$x_{t+1} = x_t^{\delta} \, \xi_t, \quad t = 0, 1, 2, ..., \tag{4}$$

where ξ_t has a log-normal distribution with mean $exp(\frac{1}{2}\sigma^2)$ and variance $exp(\sigma^2)[exp(\sigma^2) - 1]$, with initial condition $x_0 = x := e^y$.

2.1 Consumption Plan

We now present a controlled version of the previous problem (Cruz-Suárez, Montes-de-Oca, & Zacarías, 2011). First, suppose that an investor's wealth is governed by the law defined by difference equation (4), where x_t is the current wealth at time t, for $t = 0, 1, 2, ...$, $\delta < 1$, and $\{\xi_t\}$ is a sequence of independent and identically distributed random variables with density function Δ. Suppose that the investor wants to optimally manage his or her current capital x_t, dedicating part of this capital to consumption a_t and the remainder, $h(x_t) - a_t$, to investment. In particular, consider the production function $h(x) := x^{\delta}$ with $x \in \mathbb{X} := [0, \infty)$ called the *state space*. The transition law is thus given by

$$x_{t+1} \; = \; (h(x_t) - a_t) \, \xi_t \; = \; (x_t^{\delta} - a_t) \, \xi_t, \tag{5}$$

and $X_0 = x := e^y$ is known. Loans are not allowed. Therefore, $a_t \in [0, x_t^{\delta}]$ denotes consumption at time t, $\mathbb{A}(x_t) := [0, x_t^{\delta}]$ is the *admissible action space* at time t, and $\mathbb{A} := [0, +\infty)$ is the *admissible action set*.

The dynamics described in this consumption-investment system are as follows: if the system is observed at time t, the state considered is $x_t = x \in \mathbb{X} = [0, +\infty)$ and action $a_t = a \in \mathbb{A}(x)$ is applied. A reward $r(a)$ is then obtained and the system moves to the next state, $x_{t+1} \in \mathbb{X}$, by means of transition law (5). This process is repeated as discounted rewards accumulate at each time point t up to a horizon N according to a performance criterion. We denote $\mathbb{K} := \{(x, a) \,|\, x \in \mathbb{X}, a \in \mathbb{A}(x)\}$ as the set of admissible state-action pairs.

Definition 1 A consumption plan is a sequence $\pi = \{\pi_n\}_{n=0}^{\infty}$ of stochastic kernels π_n over the set of controls, given the history

$$h_n = (x, a_0, x_1, a_1 ..., x_{n-1}, a_{n-1}, x_n),$$

for each $n = 0, 1, ...$. That is, for $n \geq 0$, π_n is a stochastic kernel if it satisfies the following properties:

a) $\pi_n(\cdot \,|\, h_n)$ is a probability measure on $\mathbb{X}$, for each $h_n \in \mathbb{K} \times \mathbb{X}$.

b) $\pi_n(B \,|\, \cdot)$ is a random variable for each $B \in \mathcal{B}(\mathbb{X})$, where $\mathcal{B}(\mathbb{X})$ denotes the Borel σ-algebra of $\mathbb{X}$. The set of all plans is denoted by Π.

We thus have an initial capital $X_0 = x = e^y \in \mathbb{X}$, a plan $\pi \in \Pi$, and a utility consumption $U : \mathbb{A} \to \mathbb{R}$ defined as

$$U(a) := Log(a), \quad \forall\, a \in \mathbb{A}. \tag{6}$$

The performance criterion used to evaluate the quality of the plan $\pi \in \Pi$ is given by

$$V(\pi, x) := \mathbb{E}_x^{\pi}\Big[\sum_{t=0}^{\infty} \alpha^t\, Log(a_t)\Big], \quad \forall x \in \mathbb{X}, \tag{7}$$

where $\mathbb{E}_x^{\pi}[\cdot]$ is the expectation operator when $x \in \mathbb{X}$ is the initial condition and policy $\pi \in \Pi$ is applied. Future utilities of consumption are discounted according to a discount factor $\alpha \in (0, 1)$.

The investor's objective is to maximize the discounted consumption utility for all plans $\pi \in \Pi$, that is,

$$V(\pi^*, x) = \sup_{\pi \in \Pi} V(\pi, x) := V^*(x), \quad \forall x \in \mathbb{X}. \tag{8}$$

In this case, π^* is the optimal policy, and V^* is the optimal value function. Then, the optimal control problem consists in determining the optimal policy $\pi^* \in \Pi$.

2.2 Existence of Solution

The production function and the consumption utility function satisfy certain usual conditions, see (De La Fuente, 2000).

Let $C^2(X, Y)$ denote the set of functions $l : X \to Y$ with a continuous second derivative for Euclidean spaces X and Y.

Assumption 1 In the context of the previous problem, the production and utility functions satisfy the following conditions:

a) $h \in C^2((0, \infty), (0, \infty))$,

b) h is a concave function on $\mathbb{X}$,

c) $h' > 0$ and $h(0) = 0$,

d) $U \in C^2((0, \infty), \mathbb{R})$ is strictly increasing and strictly concave, and

e) U' is an invertible function, $U'(0) = \infty$, and $lim_{a\to\infty} U'(a) = 0$.

The following definitions will be subsequently used.

Definition 2 A measurable function $\vartheta : \mathbb{X} \to \mathbb{R}$ is said to be a solution to the optimality equation if it satisfies

$$\vartheta(x) = \sup_{a \in \mathbb{A}(x)} \left\{ Log(a) + \alpha \int_{\mathbb{X}} \vartheta(y) Q(dy|x, a) \right\}, \tag{9}$$

where Q is the corresponding transition law induced by (5) according to Section 1.2 in (Hernández-Lerma & Laserre, 1996); in other words, for $B \in \mathcal{B}(\mathbb{X})$,

$$Q(B|x, a) = \mathbb{E}_x^{\pi}\Big[\mathbb{I}_B((x^{\delta} - a)\xi)\Big], \tag{10}$$

where $\mathbb{I}_B(\cdot)$ is the indicator function of set B.

Definition 3 The value iteration functions are defined as

$$\vartheta_n(x) = \sup_{a \in \mathbb{A}(x)} \left\{ Log(a) + \alpha \int_{\mathbb{X}} \vartheta_{n-1}(y) Q(dy|x, a) \right\} \tag{11}$$

for all $x \in \mathbb{X}$ and $n = 1, 2, ...$, with $\vartheta_0(x) = 0$.

Remark 1 Under certain assumptions (Hernández-Lerma & Laserre, 1996), one can show that, for each $n = 1, 2, ...$, there exists a stationary policy $f_n \in \mathbb{F} := \{f : \mathbb{X} \to \mathbb{A} \,|\, f(x) \in \mathbb{A}(x)\}$ such that the maximum is attained in (11), that is,

$$\vartheta_n(x) = Log(f_n(x)) + \alpha \int_{\mathbb{X}} \vartheta_{n-1}(y) Q(dy|x, f_n(x)), \quad x \in \mathbb{X}.$$

Lemma 1 *For logarithmic utility and transition law (5), the following conditions are satisfied:*

a) The optimal value function V^* is a solution to the optimality equation (9).

b) For every $x \in \mathbb{X}$, $\vartheta_n(x) \to V^*(x)$ as $n \to \infty$.

Proof. See (Cruz-Suárez, Montes-de-Oca & Zacarías, 2011).

We approach the solution of the control problem with a discrete-time machine learning technique, in particular, a Q-learning method, see (Watkins & Dayan, 1992; Powell, 2011; Bertsekas, 2019). To this end, in the next section, we present a discretization procedure in state-action space.

3. Discretization of State and Action Spaces

Approximate solutions using the Q-learning method require discretization of the state and action spaces $\mathbb{X}$ and $\mathbb{A}$, respectively, and discretization requires a new set of states and actions. We therefore first consider a common finite truncated space of the original state space, namely, $I := \bar{\mathbb{X}} = [0, x]$.

We must partition the interval I into a finite number η of subintervals. We define $\rho := x/\eta$ and call ρ the grid size. Each subinterval is a subset $\mathcal{I}_i$ that consists of the nonempty intervals of the form

$$\mathcal{I}_i = (i\rho, (i+1)\rho] \cap I, \quad i = 0, 1, 2, ..., \eta.$$

We choose a representative element from each $\mathcal{I}_i$ and let $\tilde{I}$ be the set of all representative elements. For any $x \in \mathbb{X}$, we let σ_x be the element $\mathcal{I}_i$, $(i = 0, 1, 2, ..., \eta)$ to which x belongs. We also use $\tilde{\sigma}_x$ to denote the representative element of the set σ_x. For convenience, we take the right end of each subinterval $\mathcal{I}_i$ as the representative elements.

Thus, the discretized state space is given by

$$\tilde{\mathbb{X}} := \{\tilde{x}_i \in \mathbb{X} : \tilde{x}_i = i\rho \text{ for } i \in \{1, 2, ..., \eta\}\}. \tag{12}$$

For the discretization of the action space $\mathbb{A}$, we first simulate a finite number k_{max} of random values of ξ. Then, the simulated actions a_k are generated from transition law (5), such that $a_k \geq 0$ and $a_k < x^{\delta}$, because borrowing is not allowed. For simplicity, these computerized realizations are rounded. We round these values a_k and assign them to a_k^r.

Set $z_1 := Min\{a_k^r\}$ and $z_2 := Max\{a_k^r\}$, for $k \in \{1, 2, ..., k_{max}\}$, and consider $r_1 := Round\{z_1\}$ and $r_2 := Round\{z_2\}$.

Finally, we obtain the discretized admissible action space, as follows:

$$\tilde{\mathbb{A}}(\tilde{x}_i) = \left\{\tilde{a}_j \in \mathbb{A}(\tilde{x}_i) \mid \tilde{a}_j = r_1 + j/100\right\}, \quad for\ j = 0, 1, ..., (r_2 - r_1)100,\ i \in \{1, 2, ..., \eta\}.$$

Let $\hat{U} : \tilde{\mathbb{A}} \to \mathbb{R}$, be the utility function defined as

$$\hat{U}(\tilde{a}_i) := Log(\tilde{a}_i), \quad \forall\, \tilde{a}_i \in \tilde{\mathbb{A}}. \tag{13}$$

The law transition $Q(\cdot|\cdot)$ is a stochastic kernel on $\mathbb{X}$, given $\mathbb{K}$.

Define the function $\hat{\boldsymbol{Q}} : \tilde{\mathbb{X}} \times \tilde{\mathbb{A}} \times \tilde{\mathbb{X}} \to [0, 1]$ according to (Chow & Tsitsiklis, 1991) by letting

$$\hat{\boldsymbol{Q}}(\tilde{x}_j \,|\, \tilde{x}_i, \tilde{a}) := \frac{Q(\tilde{x}_j \,|\, \tilde{x}_i, a)}{\sum_{j=1}^{\eta} Q(\tilde{x}_j \,|\, \tilde{x}_i, a)}, \tag{14}$$

where Q is as in (10), that is, the probability function obtained by normalizing a lognormal density function Δ. Thus, $\{\xi_t\}$ is a sequence of independent and identically distributed random variables. Therefore,

$$\hat{\boldsymbol{Q}}(\tilde{x}_j \,|\, \tilde{x}_i, \tilde{a}) = \frac{\delta\, \tilde{x}_j^{\delta-1}}{\tilde{x}_i^{\delta} - \tilde{a}}\ \Delta\left(\frac{\tilde{x}_j^{\delta}}{\tilde{x}_i^{\delta} - \tilde{a}}\right) \frac{1}{S(\tilde{a}, \tilde{x}_i)} \tag{15}$$

for $\tilde{a} < \tilde{x}_i^{\delta}$, where

$$S(\tilde{a}, \tilde{x}_i) = \sum_{j=1}^{\eta} \frac{\delta\, \tilde{x}_j^{\delta-1}}{\tilde{x}_i^{\delta} - \tilde{a}}\ \Delta\left(\frac{\tilde{x}_j^{\delta}}{\tilde{x}_i^{\delta} - \tilde{a}}\right),$$

which is interpreted as the probability of the transition to state $\tilde{x}_j$ when the current state is $\tilde{x}_i$ and action $\tilde{a}$ is chosen.

4. Q-learning

4.1 Development of the Q-Learning Method

Bellman's optimality equation, see (Gosavi, 2015) yields

$$J^*(\tilde{x}_i) := \max_{\tilde{a}\in\tilde{\mathbb{A}}(\tilde{x}_i)} \left\{ Log(\tilde{a}) + \alpha \sum_{j=1}^{\eta} J^*(\tilde{x}_j)\, \hat{\mathbf{Q}}(\tilde{x}_j \,|\, \tilde{x}_i, \tilde{a}) \right\} \tag{16}$$

for all $\tilde{x}_i \in \tilde{\mathbb{X}}$, where $J^*(\tilde{x}_i)$ denotes the ith element of the objective function vector associated with the optimal policy.

Definition 4 For $\tilde{x}_i \in \tilde{\mathbb{X}}$, $\tilde{a} \in \tilde{\mathbb{A}}(\tilde{x}_i)$ we define the *Q-factors* function as follows:

$$Q(\tilde{x}_i, \tilde{a}) := Log(\tilde{a}) + \alpha \sum_{j=1}^{\eta} J^*(\tilde{x}_j)\, \hat{\mathbf{Q}}(\tilde{x}_j \,|\, \tilde{x}_i, \tilde{a}). \tag{17}$$

Then,

$$J^*(\tilde{x}_i) = \max_{\tilde{a}\in\tilde{\mathbb{A}}(\tilde{x}_i)} Q(\tilde{x}_i, \tilde{a}).$$

Therefore, we can write the following equation which is known as the version of the *Bellman optimality equation for Q-factors*:

$$Q(\tilde{x}_i, \tilde{a}) = Log(\tilde{a}) + \alpha \sum_{j=1}^{\eta} \hat{\mathbf{Q}}(\tilde{x}_j \,|\, \tilde{x}_i, \tilde{a}) \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q(\tilde{x}_j, \tilde{b}) \tag{18}$$

for $\tilde{x}_i \in \tilde{\mathbb{X}}$, $\tilde{a} \in \tilde{\mathbb{A}}(\tilde{x}_i)$.

4.2 Value Iteration for Q-factors

Our value iteration algorithm, Algorithm 1, is a classic approximation algorithm that has been applied to Q-factors, see (Bertsekas, 2005; Almudevar, 2014; Gosavi, 2015).

Notation The iterated approximations of the optimal policy $\pi^* \in \Pi$ generated by the Q-learning method for each $\tilde{x} \in \tilde{\mathbb{X}}$ are denoted by $\tilde{\pi}_k(\tilde{x})$, $k = 0, 1, 2, \ldots$.

Algorithm 1: Value iteration for Q-factors

Initialization $k = 0$, specify $\epsilon > 0$ (or define k_{max} number of iterations), $Q_0(\tilde{x}_i, \tilde{a}) = 0$ for each $\tilde{x}_i \in \tilde{\mathbb{X}}$;
repeat
  for $\tilde{x}_i \in \tilde{\mathbb{X}}$ *and* $\tilde{a} \in \tilde{\mathbb{A}}(\tilde{x}_i)$ **do**
    $Q_{k+1}(\tilde{x}_i, \tilde{a}) \longleftarrow Log(\tilde{a}) + \alpha \sum_{j=1}^{\eta} \hat{\mathbf{Q}}(\tilde{x}_j \,|\, \tilde{x}_i, \tilde{a}) \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q_k(\tilde{x}_j, \tilde{b})$;
    $J^{k+1}(\tilde{x}_i) \longleftarrow \max_{\tilde{a}\in\tilde{\mathbb{A}}(\tilde{x}_i)} Q_{k+1}(\tilde{x}_i, \tilde{a})$;
    $J^{k}(\tilde{x}_i) \longleftarrow \max_{\tilde{a}\in\tilde{\mathbb{A}}(\tilde{x}_i)} Q_{k}(\tilde{x}_i, \tilde{a})$;
  end
  $k \longleftarrow k + 1$;
until $\|J^{k+1} - J^k\| < \epsilon(1-\alpha)/2\alpha$, *(or if* $k \geq k_{max}$*)*;
return $\tilde{\pi}^*(\tilde{x}_i) \in \underset{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_i)}{argmax}\, Q_\epsilon(\tilde{x}_i, \tilde{b})$, *(or* $Q_{k_{max}}(\tilde{x}_i, \tilde{b})$*).*

4.3 Robbins-Monro Algorithm for Q-Factors

Recalling the Q-Factor definition in the form of the Bellman equation, we have

$$\begin{aligned} Q(\tilde{x}_i, \tilde{a}) &= Log(\tilde{a}) + \alpha \sum_{j=1}^{\eta} \hat{\mathbf{Q}}(\tilde{x}_j \,|\, \tilde{x}_i, \tilde{a}) \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q(\tilde{x}_j, \tilde{b}) \\ &= \sum_{j=1}^{\eta} \hat{\mathbf{Q}}(\tilde{x}_j \,|\, \tilde{x}_i, \tilde{a}) \Big[Log(\tilde{a}) + \alpha \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q(\tilde{x}_j, \tilde{b}) \Big] \\ &= \mathbb{E}\Big[Log(\tilde{a}) + \alpha \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q(\tilde{x}_j, \tilde{b}) \Big]. \end{aligned}$$

The expectation can be estimated using the Monte Carlo method (Ross, 2013), for instance; that is, it is possible to estimate the Q-factors by using the Robbins-Monro algorithm scheme, see (Robbins & Monro, 1951):

$$\begin{aligned} Q_{k+1}(\tilde{x}_i, \tilde{a}) &= (1-\lambda_{k+1})Q_k(\tilde{x}_i, \tilde{a}) + \lambda_{k+1}\Big[Log(\tilde{a}) + \alpha \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q_k(\tilde{x}_j, \tilde{b})\Big] \\ &= Q_k(\tilde{x}_i, \tilde{a}) + \lambda_{k+1}\Big[Log(\tilde{a}) + \alpha \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q_k(\tilde{x}_j, \tilde{b}) - Q_k(\tilde{x}_i, \tilde{a})\Big], \end{aligned}$$

where the function λ_k is the *reinforcement* or *learning rate*. We derive $\lambda_k = 1/(k+1)$ from the Robbins-Monro algorithm. Other reinforcement rates can also be considered, as long as they satisfy the conditions:

$$\sum_{k=1}^{\infty} \lambda_k = \infty \quad and \quad \sum_{k=1}^{\infty} (\lambda_k)^2 < \infty.$$

4.4 Q-learning Modified Algorithm

The algorithm for applying Q-learning to Markov decision processes with a discounted reward through value iteration is presented as Algorithm 2.

Algorithm 2: Q-learning modified algorithm

Initialization Specify k_{max} and parameters of $\{\xi\}$;
for $k = 0$ **to** k_{max} **do**
  Simulate ξ_k;
  $a_k \longleftarrow x^{\delta} - \frac{x}{\xi_k}$ such that $a_k \geq 0$ and $a_k < x^{\delta}$;
  $a_k^r \longleftarrow$ Round$[a_k]$;
end
Set $r_1 := Round\{Min\{a_k^r\}\}$, $r_2 := Round\{Max\{a_k^r\}\}$;
for $j = 0$ **to** $(r_2 - r_1)100$ **do**
  $\tilde{a}_j \longleftarrow r_1 + j/100 \quad (\tilde{a}_j \in \tilde{\mathbb{A}}(\tilde{x}))$;
  $Q_0(\tilde{x}, \tilde{a}_j) \longleftarrow 0$;
end
repeat
  for $x = \tilde{x}_i \in \tilde{\mathbb{X}}$ *and* $\tilde{a} \in \tilde{\mathbb{A}}(\tilde{x}_i)$ **do**
    $Q_{k+1}(\tilde{x}_i, \tilde{a}_j) \longleftarrow Q_k(\tilde{x}_i, \tilde{a}_j) + \lambda_k\Big[Log(\tilde{a}_j) + \alpha \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q_k(\tilde{x}_j, \tilde{b}) - Q_k(\tilde{x}_i, \tilde{a}_j)\Big]$;
  end
  Simulate the next state $\tilde{x}_j$, until returning to $x = \tilde{x}_i$;
  $k \longleftarrow k + 1$;
until $k \geq k_{max}$;
return $\tilde{\pi}^*(x) \in \underset{\tilde{b}\in\tilde{\mathbb{A}}(x)}{argmax}\ Q_{k_{max}}(x, \tilde{b})$.

4.5 Q-learning Convergence

Definition 5 The asynchronous stochastic approximation scheme is given by the *recursive formula of the Q-learning method*:

$$Q_{k+1}(\tilde{x}_i, \tilde{a}) = Q_k(\tilde{x}_i, \tilde{a}) + \lambda_k \Big[Log(\tilde{a}) + \alpha \max_{\tilde{b}\in\tilde{\mathbb{A}}(\tilde{x}_j)} Q_k(\tilde{x}_j, \tilde{b}) - Q_k(\tilde{x}_i, \tilde{a})\Big], \tag{19}$$

for $k = 1, 2, ..., k_{max}$, where k_{max} is the number of iterations.

Note that the following statement holds.

Remark 2 Let Θ^k be the index of the state-action pair visited in the kth iteration of the algorithm, and $\mathbb{I}(\cdot)$ the indicator function. We then define

$$V_k(\tilde{x}_i, \tilde{a}) := \sum_{m=1}^{k} \mathbb{I}\Big((\tilde{x}_i, \tilde{a}) = \Theta^k\Big), \quad for\ \tilde{x}_i \in \tilde{\mathbb{X}}\ and\ \tilde{a} \in [0, \tilde{x}_i^{\delta}],$$

that is, the indicator function will return a zero for a Q-factor that is not visited in the kth iteration, which implies that Q-factors that are not visited in that iteration will not be updated. There is a deterministic scalar $\chi > 0$ such that we almost surely obtain for all $\tilde{x}_i \in C := \tilde{\mathbb{X}} \backslash \{0\}$ and $\tilde{a} \in [0, \tilde{x}_i^{\delta}]$,

$$\liminf_{k\to\infty} \frac{V_k(\tilde{x}_i, \tilde{a})}{k} \geq \chi,$$

because C is an irreducible closed set of recurrent states. Moreover, when

$$K^k(z) := min\left\{ m > k \ : \ \sum_{s=k+1}^{m} \frac{Log(s)}{s} > z \right\},$$

we obtain for any $z > 0$, the limit

$$\lim_{k\to\infty} \frac{\sum_{m=V_k(l)}^{V_{K^k(z)}(l)} \frac{Log(m)}{m}}{\sum_{m=V_k(l')}^{V_{K^k(z)}(l')} \frac{Log(m)}{m}} < +\infty$$

almost surely.

Proposition 1 *The step size $\lambda_k(\tilde{x}_i, \tilde{a}) = \frac{Log(k)}{k}$ and the action selected in the Q-learning algorithm satisfy the following statements:*

a) *The step size $\lambda_k(\tilde{x}_i, \tilde{a})$ satisfies the following conditions for all $\tilde{x}_i \in \tilde{\mathbb{X}}$ and $\tilde{a} \in [0, \tilde{x}_i^{\delta}]$:*

$$\sum_{k=1}^{\infty} \lambda_k(\tilde{x}_i, \tilde{a}) = \infty, \qquad \sum_{k=1}^{\infty} \lambda_k^2(\tilde{x}_i, \tilde{a}) < \infty.$$

b) *For all $\tilde{x}_i \in \tilde{\mathbb{X}}$ and $\tilde{a} \in [0, \tilde{x}_i^{\delta}]$*

i) $\lambda_{k+1}(\tilde{x}_i, \tilde{a}) \leq \lambda_k(\tilde{x}_i, \tilde{a})$ for some value of k onward.

ii) For any $z \in (0, 1)$, $sup_k\, \lambda_{[zk]}(\tilde{x}_i, \tilde{a})/\lambda_k(\tilde{x}_i, \tilde{a}) < \infty$.

iii) For any $z \in (0, 1)$,

$$\lim_{k\to\infty} \frac{\sum_{m=1}^{[zk]+1} \lambda_m(\tilde{x}_i, \tilde{a})}{\sum_{m=1}^{k} \lambda_m(\tilde{x}_i, \tilde{a})} = 1.$$

Proof. Consider $\lambda_k = \frac{Log(k)}{k}$, $k = 1, 2, ...$

a) Note that

$$\limsup_{k\to\infty} \frac{k}{k+1} \frac{Log(k+1)}{Log(k)} > \limsup_{k\to\infty} \frac{k}{k+1} = 1, \tag{20}$$

and, therefore,

$$\sum_{k=1}^{\infty} \frac{Log(k)}{k} = +\infty.$$

Now, applying the properties of the zeta function, ζ, we can demonstrate that

$$\sum_{k=1}^{\infty} \left(\frac{Log(k)}{k} \right)^2 = \zeta''(2) \approx 1.98. \tag{21}$$

Therefore, the result follows, given (20) and (21).

b) Moreover, λ_k satisfies the following properties:

i) λ_k is strictly decreasing, $\forall k \geq 1$.

ii) Moreover, for $z \in (0, 1)$, $\sup_k \lambda_{[zk]}/\lambda_k = \sup_k \frac{kLog([zk])}{[zk]Log(k)} < \infty$.

iii) Additionally, if $z \in (0, 1)$, we have

$$\lim_{k\to\infty} \frac{\sum_{m=1}^{[zk]+1} \frac{Log(m)}{m}}{\sum_{m=1}^{k} \frac{Log(m)}{m}} = 1.$$

The following proposition is a consequence of Remark 2 and Proposition 1, see (Gosavi, 2015).

Proposition 2 The sequence of policies generated by the Q-learning algorithm, $\{\tilde{\pi}_k\}_{k=1}^{\infty}$, converges to π^*.

5. Numerical Example

To exemplify Algorithm 2, we consider the performance criteria to be optimized, as follows:

$$V(\pi, x) := \mathbb{E}_x^{\pi}\left[\sum_{t=1}^{\infty} \alpha^t \, Log(\tilde{a}_t)\right], \quad \pi \in \Pi, x \in \mathbb{X},$$

where the initial capital is x and $\alpha = 0.5$ is the discount factor.

The transition of the system is governed by (5), where $\delta = 0.75$ and $\{\xi_t\}$ is such that $Log(\xi_t) \sim N(0, 0.5)$, which leads to the normalized mass probability function on space $\tilde{\mathbb{X}} \times \tilde{\mathbb{A}} \times \tilde{\mathbb{X}}$ given by (15).

For the Q-learning method, set the step size $\lambda_k(\tilde{x}_i, \tilde{a}) := \frac{Log(k)}{k}$ as the reinforcement rate and $k_{max} = 1000$. Table 1 and Figure 1 summarize some results, compared with the exact solution.

(Cruz-Suárez, Montes-de-Oca, & Zacarías, 2011) find that the optimal policy is given by

$$a^*(x) = x^{\delta}(1 - \alpha\delta), \quad x \in \mathbb{X}.$$

Table 1. Optimal policy

x	$\tilde{a}^*(x)$	$a^*(x)$
1.0	0.64	0.625
2.0	1.12	1.051
3.0	1.30	1.424
4.0	1.68	1.767
5.0	2.10	2.089

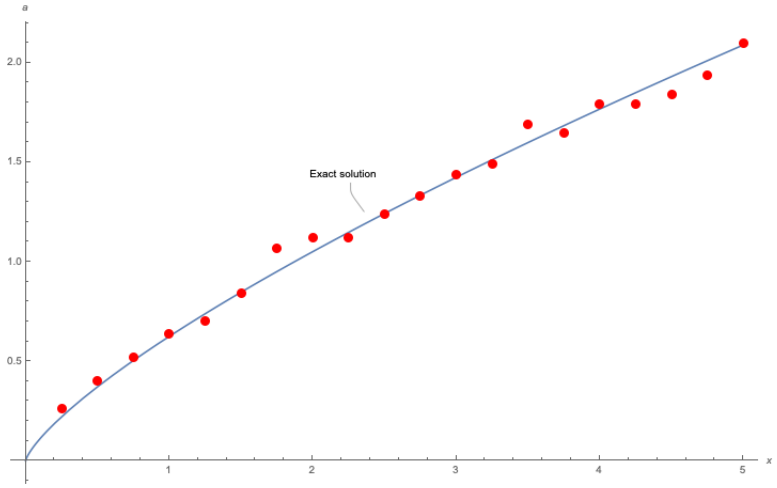

Figure 1. Approximate optimal policy

Table 1 shows approximate optimal actions for some of the values of x, obtained using the method described in Algorithm 2, contrasted with the exact optimal policy. Figure 1 shows further approximate optimal actions, denoted by red dots, which exhibit a behavior similar to that of the exact optimal policy, graphed by the solid blue line.

The proposed modified Q-Learning algorithm is a good approximation for the optimal actions with the given parameters without the need to resort to a closed solution, if there is one.

6. Conclusion

The method proposed in this work offers an alternative source of solutions to the optimal policy of a control problem. This method is also useful when a closed solution is not available or when classical resolution tools fail to find the optimal policy directly.

In the optimal investment strategies that are derived in closed form, the candidate functions are usually proposed with the general form of the solution. These functions can be very difficult to find when using more complex or unknown utility or production functions.

Reinforcement learning methods have an advantage in terms of implementation capacity, assigning values of goodness to state-action pairs, putting them to the test, and discarding action selections that produce adverse results.

References

Almudevar, A. (2014). *Approximate Iterative Algorithms*. London: CRC Press/Balkema. https://doi.org/10.1201/b16613

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*. Nashua, NH: Athena Scientific.

Bertsekas, D. P. (2019). *Reinforcement Learning and Optimal Control*. Nashua, NH: Athena Scientific.

Chow, C. S., & Tsitsiklis, J. N. (1991). An optimal one-way multigrid algorithm for discrete-time stochastic control. *IEEE Transactions on Optimal Control, 36*(8), 898-914. https://doi.org/10.1109/9.133184

Cruz-Suárez, H. A., Montes-de-Oca, R., & Zacarías, G. (2011). A consumption-investment problem modelled as a discounted Markov decision process, *Kybernetika, 47*(6), 909-929.

De La Fuente, A. (2000). *Mathematical Methods and Models*. Cambridge: Cambridge University Press.

Gosavi, A. (2015). *Simulation-Based Optimization Parametric Optimization Techniques and Reinforcement Learning*. New York: Springer. https://doi.org/10.1007/978-1-4899-7491-4_7

Hernández-Lerma, O., & Laserre, J. B. (1996). *Discrete-Time Markov Control Processes Basic Optimality Criteria*. New York: Springer. https://doi.org/10.1007/978-1-4612-0729-0

Higham, D. (2001). An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review, 43*(3), 525-546. https://doi.org/10.1137/S0036144500378302

Liang, Z., & Ma, M. (2019). ConsumptionCinvestment problem with pathwise ambiguity under logarithmic utility. *Mathematics and Financial Economics, 13*(4), 519-541. https://doi.org/10.1007/s11579-019-00236-y

Mikosch, T. (1998). *Elementary Stochastic Calculus with Finance View*. Advanced Series on Statistical Science & Applied Probability, Vol. 6. Singapore: World Scientific. https://doi.org/10.1142/3856

Mendelssohn, R., & Sobel, M. (1980). Capital accumulation and the optimization of renewable resource models, *Journal of Economic Theory, 23*(2), 243-260. https://doi.org/10.1016/0022-0531(80)90009-5

Powell, W. (2011). *Approximate Dynamic Programming. Solving the Curses of Dimensionality.* Hoboken, NJ: John Wiley. https://doi.org/10.1002/9781118029176

Robbins, H., & Monro, S. (1951). A stochastic approximation method, *Annals of Mathematical Statistics, 22*(3), 400-407. https://doi.org/10.1214/aoms/1177729586

Ross, S. M. (2013). *Simulation*. San Diego, CA: Academic Press.

Samuelson, P. (1969). Lifetime portfolio selection by dynamic stochastic programming. *Review of Economics and Statistics, 51*(3), 239-246. https://doi.org/10.2307/1926559

Singh, S. (1994). Learning to solve Markovian decision processes. PhD thesis, University of Massachusetts.

Stokey, N., & Lucas, R. (1989). *Recursive Methods in Economic Dynamics* Cambridge, MA: Harvard University Press. https://doi.org/10.2307/j.ctvjnrt76

Sutton, R., & Barto, A. (2018) *Reinforcement Learning. An Introduction*, Cambridge, MA: MIT Press.

Tsitsiklis, J. (1994). Asynchronous stochastic approximation and Q-learning. *Machine Learning, 16*, 185-202. https://doi.org/10.1007/BF00993306

Vitoriano, B., & Parlier, G. H. (Eds.). (2017). Operations Research and Enterprise Systems, Fifth International Conference, ICORES 2016, Rome, Italy, February 23-25, 2016, Revised Selected Papers, Vol. 695. Rome: Springer. https://doi.org/10.1007/978-3-319-53982-9

Watkins, C. (1989). Learning from delayed rewards. PhD thesis. Cambridge, King's College.

Watkins, C., & Dayan, P. (1992). Q-Learning. Technical note. *Machine Learning*, (8), pp. 279-292. Boston, MA: Kluwer Academic Publishers. https://doi.org/10.1007/BF00992698

Weissensteiner, A. (2009). A *Q*-Learning Approach to Derive Optimal Consumption and Investment Strategies. *IEEE*

transactions on neural networks, 20(8), 1234-1243. https://doi.org/10.1109/TNN.2009.2020850

White, D. (1993). A survey of applications of Markov decision processes. *Journal of the Operational Research Society, 44*, 1073-1096. https://doi.org/10.2307/2583870