CHAPTER 12

Reinforcement Learning in Finance

Gordon Ritter

12.1 INTRODUCTION

We live in a period characterized by rapid advances in artificial intelligence (AI) and machine learning, which are transforming everyday life in amazing ways. AlphaGo Zero (Silver et al. 2017) showed that superhuman performance can be achieved by pure reinforcement learning, with only very minimal domain knowledge and (amazingly!) *no reliance on human data or guidance*. AlphaGo Zero learned to play after merely being told the rules of the game, and playing against a simulator (itself, in that case).

The game of Go has many aspects in common with trading. Good traders often use complex strategy and plan several periods ahead. They sometimes make decisions which are 'long-term greedy' and pay the cost of a short-term temporary loss in order to implement their long-term plan. In each instant, there is a relatively small, discrete set of actions that the agent can take. In games such as Go and chess, the available actions are dictated by the rules of the game.

In trading, there are also rules of the game. Currently the most widely used trading mechanism in financial markets is the 'continuous double auction electronic order book with time priority'. With this mechanism, quote arrival and transactions are continuous in time and execution priority is assigned based on the price of quotes and their arrival order. When a buy (respectively, sell) order x is submitted, the exchange's matching engine checks whether it is possible to match x to some other previously submitted sell (respectively, buy) order. If so, the matching occurs immediately. If not, x becomes *active* and it remains active until either it becomes matched to another incoming sell (respectively, buy) order or it is cancelled. The set of all active orders at a given price level is a FIFO queue. There's much more to say about microstructure theory and we refer to the book by Hasbrouck (2007), but these are, in a nutshell, the 'rules of the game'.

These observations suggest the inception of a new subfield of quantitative finance: reinforcement learning for trading (Ritter 2017). Perhaps the most fundamental question in this burgeoning new field is the following.

Big Data and Machine Learning in Quantitative Investment, First Edition.
Tony Guida.

FUNDAMENTAL QUESTION 1

Can an artificial intelligence discover an *optimal* dynamic trading strategy (with transaction costs) without being told what kind of strategy to look for?

If it existed, this would be the financial analogue of AlphaGo Zero.

In this note, we treat the various elements of this question:

1. What is an optimal dynamic trading strategy? How do we calculate its costs?
2. Which learning methods have even a chance at attacking such a difficult problem?
3. How can we engineer the reward function so that an AI has the potential to learn to optimize the right thing?

The first of these sub-problems is perhaps the easiest. In finance, *optimal* means that the strategy optimizes expected utility of final wealth (utility is a subtle concept and will be explained below). Final wealth is the sum of initial wealth plus a number of wealth increments over shorter time periods:

$$\text{maximize}: \mathbb{E}[u(w_T)] = \mathbb{E}\left[u\left(w_0 + \sum_{t=1}^{T} \delta w_t\right)\right] \tag{12.1}$$

where $\delta w_t = w_t - w_{t-1}$. Costs include market impact, crossing the bid-offer spread, commissions, borrow costs, etc. These costs generally induce a negative drag on w_T because each δw_t is reduced by the cost paid in that period.

We now discuss the second question: which learning methods have a chance of working? When a child tries to ride a bicycle (without training wheels) for the first time, the child cannot perfectly know the exact sequence of actions (pedal, turn handlebars, lean left or right, etc.) that will result in the bicycle remaining balanced and going forward. There is a trial and error process in which the correct actions are rewarded; incorrect actions incur a penalty. One needs a coherent mathematical framework which promises to mimic or capture this aspect of how sentient beings learn.

Moreover, sophisticated agents/actors/beings are capable of complex strategic planning. This usually involves thinking several periods ahead and perhaps taking a small loss to achieve a greater anticipated gain in subsequent periods. An obvious example is losing a pawn in chess as part of a multi-part strategy to capture the opponent's queen. How do we teach machines to 'think strategically'?

Many intelligent actions are deemed 'intelligent' precisely because they are optimal interactions with an environment. An algorithm plays a computer game intelligently if it can optimize the score. A robot navigates intelligently if it finds a shortest path with no collisions: minimizing a function which entails path length with a large negative penalty for collision.

Learning, in this context, is learning how to choose actions wisely to optimize your interaction with your environment, in such a way as to maximize rewards received over time. Within artificial intelligence, the subfield dedicated to the study of this kind of learning is called *reinforcement learning*. Most of its key developments are summarized in Sutton and Barto (2018).

An oft-quoted adage is that there are essentially three types of machine learning: supervised, unsupervised and reinforcement. As the story goes, *supervised learning* is

learning from a labelled set of examples called a 'training set' while *unsupervised learning* is finding structure hidden in collections of unlabelled data, and reinforcement learning is something else entirely. The reality is that these forms of learning are all interconnected. Most production-quality reinforcement learning systems employ elements of supervised and unsupervised learning as part of the representation of the value function. Reinforcement learning is about maximizing cumulative reward over time, not about finding hidden structure, but it is often the case that the best way to maximize the reward signal is by finding hidden structure.

12.2 MARKOV DECISION PROCESSES: A GENERAL FRAMEWORK FOR DECISION MAKING

Sutton and Barto (1998) say:

> The key idea of reinforcement learning generally, is the use of value functions to organize and structure the search for good policies.

The foundational treatise on value functions was written by Bellman (1957), at a time when the phrase 'machine learning' was not in common usage. Nonetheless, reinforcement learning owes its existence, in part, to Richard Bellman.

A *value function* is a mathematical expectation in a certain probability space. The underlying probability measure is the one associated to a system which is very familiar to classically trained statisticians: a Markov process. When the Markov process describes the state of a system, it is sometimes called a *state-space model*. When, on top of a Markov process, you have the possibility of choosing a *decision* (or action) from a menu of available possibilities (the 'action space'), with some reward metric that tells you how good your choices were, then it is called a *Markov decision process* (MDP).

In a Markov decision process, once we observe the current state of the system, we have the information we need to make a decision. In other words (assuming we know the current state), then it would not help us (i.e. we could not make a better decision) to also know the full history of past states which led to the current state. This history-dependence is closely related to Bellman's principle.

Bellman (1957) writes: 'In each process, the functional equation governing the process was obtained by an application of the following intuitive.'

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.
>
> Bellman (1957)

The 'functional equations' that Bellman is talking about are essentially (12.7) and (12.8), as we explain in the next section. Consider an interacting system: agent interacts with environment. The 'environment' is the part of the system outside of the agent's *direct* control. At each time-step t, the agent observes the current state of the environment $S_t \in \mathrm{S}$ and chooses an action $A_t \in \mathrm{A}$. This choice influences both the transition to the next state and the reward the agent receives (Figure 12.1).

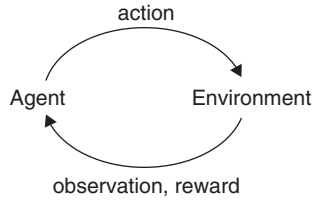

FIGURE 12.1 Interacting system: agent interacts with environment.

Underlying everything, there is assumed to be a distribution

$$p\left(s^0, r \mid s, a\right)$$

for the joint probability of transitioning to state $s^0 \in \mathrm{S}$ and receiving reward r, conditional on the previous state being s and the agent taking action a. This distribution is typically not known to the agent, but its existence gives mathematical meaning to notions such as 'expected reward'.

The agent's goal is to maximize the expected cumulative reward, denoted by

$$Gt = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \ldots \tag{12.2}$$

where $0 < \gamma < 1$ is necessary for the infinite sum to be defined.

A *policy* π is, roughly, an algorithm for choosing the next action, based on the state you are in. More formally, a policy is a mapping from states to probability distributions over the action space. If the agent is following policy π, then in state s, the agent will choose action a with probability $\pi(a|s)$.

Reinforcement learning is the search for policies which maximize

$$\mathbb{E}[G_t] = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\right]$$

Normally, the policy space is too large to allow brute-force search, so the search for policies possessing good properties must proceed by the use of value functions.

The *state-value function* for policy π is defined to be

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

where $\mathrm{E}\pi$ denotes the expectation under the assumption that policy π is followed. For any policy π and any state s, the following consistency condition holds:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t \mid S_t = s] && (12.3)\\ &= \mathbb{E}_\pi[\mathrm{R}_{t+1} + \gamma G_{t+1}\} S_t = s] && (12.4)\\ &= \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma E_\pi[G_{t+1} \mid S_{t+1} - s']] && (12.5)\\ &= \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')] && (12.6) \end{aligned}$$

The end result of the above calculation,

$$v_\pi(s) = \sum_{a,s',r} \pi(a \mid s)p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$

is called the *Bellman equation*. The value function v_π is the unique solution to its Bellman equation. Similarly, the *action-value function* expresses the value of starting in state s, taking action a, and then following policy π thereafter:

$$q_\pi(s, a) := \mathrm{E}\pi[G_t \mid S_t = s, A_t = a]$$

Policy π is defined to be at least as good as π^0 if

$$v_\pi(s) \geq v_\pi 0(s)$$

for all states s.

An *optimal policy* is defined to be one which is at least as good as any other policy. There need not be a unique optimal policy, but all optimal policies share the same optimal state-value function

$$v_*(s) = \max_\pi v_\pi(s)$$

and optimal action-value function

$$q_*(s, a) = \max_\pi q_\pi(s, a).$$

Note that $v_*(s) = \max_a q_*(s,a)$, so the action-value function is more general than the state-value function.

The optimal state-value function and action-value function satisfy the Bellman equations

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_*(s')] \tag{12.7}$$

$$q_*(s, a) = \sum_{s',r} p(s', r \mid s, a)[r + \gamma \max_{a'} q_*(s', a)] \tag{12.8}$$

where the sum over s^0,r denotes a sum over all states s^0 and all rewards r.

If we possess a function $q(s,a)$ which is an estimate of $q_*(s,a)$, then the *greedy policy* (associated to the function q) is defined as picking at time t the action a_t^* which maximizes $q(s_t,a)$ over all possible a, where s_t is the state at time t. Given the function q_*, the associated greedy policy is the optimal policy. Hence we can reduce the problem to finding q_*, or producing a sequence of iterates that converges to q_*.

It is worth noting at this point that modern approaches to multi-period portfolio optimization with transaction costs (Gârleanu and Pedersen 2013; Kolm and Ritter 2015; Benveniste and Ritter 2017) are also organized as optimal control problems which, in principle, could be approached by finding solutions to (12.7), although these equations are difficult to solve with constraints and non-differentiable costs.

There is a simple algorithm which produces a sequence of functions converging to q_*, known as Q-learning (Watkins 1989). Many subsequent advancements built upon

Watkins' seminal work, and the original form of Q-learning is perhaps no longer the state of the art. Among its drawbacks include that it can require a large number of time-steps for convergence.

The Watkins algorithm consists of the following steps. One initializes a matrix Q with one row per state and one column per action. This matrix can be initially the zero matrix, or initialized with some prior information if available. Let S denote the current state.

Repeat the following steps until a pre-selected convergence criterion is obtained:

1. Choose action $A \in \mathrm{A}$ using a policy derived from Q which combines exploration and exploitation.
2. Take action A, after which the new state of the environment is S^0 and we observe reward R.
3. Update the value of $Q(S,A)$: set.

$$\text{Target} = R + \gamma \max_a Q(S', a)$$

and

$$Q(S, A) += \alpha \underbrace{[\text{Target} - Q(S, A)]}_{\text{TD-error}} \tag{12.9}$$

where $\alpha \in (0,1)$ is called the step-size parameter. The step-size parameter does not have to be constant and indeed can vary with each time-step. Convergence proofs usually require this; presumably the MDP generating the rewards has some unavoidable process noise which cannot be removed with better learning, and thus the variance of the TD-error never goes to zero. Hence α_t must go to zero for large time-step t.

Assuming that all state-action pairs continue to be updated, and assuming a variant of the usual stochastic approximation conditions on the sequence of step-size parameters (see (12.10) below), the Q-learning algorithm has been shown to converge with probability 1 to q_*.

In many problems of interest, either the state space, or the action space, or both are most naturally modelled as continuous spaces (i.e. sub-spaces of R^d for some appropriate dimension d). In such a case, one cannot directly use the algorithm above. Moreover, the convergence result stated above does not generalize in any obvious way.

Many recent studies (for example, Mnih et al. (2015)) have proposed replacing the Q-matrix in the above algorithm with a deep neural network. Unlike a lookup table, a neural network has a vector of associated parameters, sometimes called *weights*. We could then write the Q-function as $Q(s,a;\theta)$, emphasizing its parameter dependence. Instead of iteratively updating values in a table, we will iteratively update the parameters, θ, so that the network learns to compute better estimates of state-action values.

Although neural networks are general function approximators, depending on the structure of the truly-optimal Q-function q_*, it may be the case that a neural network learns very slowly, both in terms of the CPU time needed to train and also in terms of efficient sample use (defined as a lower number of samples required to achieve acceptable results). The network topology and the various choices such as activation functions and optimizer (Kingma and Ba 2014) matter quite a bit in terms of training time and efficient sample use.

In some problems, the use of simpler function-approximators (such as ensembles of regression trees) to represent the unknown function $Q(s,a;\theta)$ may lead to much faster training time and much more efficient sample use. This is especially true when q_* can be well approximated by a simple functional form such as a locally linear form.

The observant reader will have noticed the similarity of the Q-learning update procedure to stochastic gradient descent. This fruitful connection was noted and exploited by Baird III and Moore (1999), who reformulate several different learning procedures as special cases of stochastic gradient descent.

The convergence of stochastic gradient descent has been studied extensively in the stochastic approximation literature (Bottou 2012). Convergence results typically require learning rates satisfying the conditions

$$\sum_t \alpha_t^2 < \infty \text{ and } \sum_t \alpha_t = \infty \tag{12.10}$$

The theorem of Robbins and Siegmund (1985) provides a means to establish almost sure convergence of stochastic gradient descent under surprisingly mild conditions, including cases where the loss function is non-smooth.

12.3 RATIONALITY AND DECISION MAKING UNDER UNCERTAINTY

Given some set of mutually exclusive outcomes (each of which presumably affects wealth or consumption somehow), a *lottery* is a probability distribution on these events such that the total probability is 1. Often, but not always, these outcomes involve gain or loss of wealth. For example, 'pay 1000 for a 20% chance to win 10,000' is a lottery.

Nicolas Bernoulli described the St Petersburg lottery/paradox in a letter to Pierre Raymond de Montmort on 9 September 1713. A casino offers a game of chance for a single player in which a fair coin is tossed at each stage. The pot starts at \$2 and is doubled every time a head appears. The first time a tail appears, the game ends and the player wins whatever is in the pot. The mathematical expectation value is

$$\frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 4 + \cdots = +\infty$$

This paradox led to a remarkable number of new developments as mathematicians and economists struggled to understand all the ways it could be resolved.

Daniel Bernoulli (cousin of Nicolas) in 1738 published a seminal treatise in the Commentaries of the Imperial Academy of Science of Saint Petersburg; this is, in fact, where the modern name of this paradox originates. Bernoulli's work possesses a modern translation (Bernoulli 1954) and thus may be appreciated by the English-speaking world.

> The determination of the value of an item must not be based on the price, but rather on the utility it yields… There is no doubt that a gain of one thousand ducats is more significant to the pauper than to a rich man though both gain the same amount.
>
> Bernoulli (1954)

Among other things, Bernoulli's paper contains a proposed resolution to the paradox: if investors have a logarithmic utility, then their expected change in utility of wealth by playing the game is finite.

Of course, if our only goal were to study the St Petersburg paradox, then there are other, more practical resolutions. In the St Petersburg lottery only very unlikely events yield the high prizes that lead to an infinite expected value, so the expected value becomes finite if we are willing to, as a practical matter, disregard events which are expected to occur less than once in the entire lifetime of the universe.

Moreover, the expected value of the lottery, even when played against a casino with the largest resources realistically conceivable, is quite modest. If the total resources (or total maximum jackpot) of the casino are W dollars, then $L = \text{blog}_2(W)\text{c}$ is the maximum number of times the casino can play before it no longer fully covers the next bet, i.e. $2^L \leq W$ but $2^{L+1} > W$. The mutually exclusive events are that you flip one time, two times, three times, ..., L times, winning $2^1, 2^2, 2^3, \ldots, 2^L$ or finally, you flip 2^{L+1} times and win W. The expected value of the lottery then becomes:

$$\sum_{k=1}^{L} \frac{1}{2^k} \cdot 2^k + \left(1 - \sum_{k=1}^{L} \frac{1}{2^k}\right) W = L + W2^{-L}$$

If the casino has $W = \$1$ billion, the expected value of the 'realistic St Petersburg lottery' is only about \$30.86.

If lottery M is preferred over lottery L, we write $L \prec M$. If M is either preferred over or viewed with indifference relative to L, we write $L \preceq M$. If the agent is indifferent between L and M, we write $L \sim M$. Von Neumann and Morgenstern (1945) provided the definition of when the preference relation is rational, and proved the key result that any rational preference relation has an expression in terms of a utility function.

Definition 1 (Von Neumann and Morgenstern 1945). A preference relation is said to be *rational* if all of the four axioms hold:

1. For any lotteries L,M, exactly one of the following holds:

$$L \prec M, M \prec L, or\, L \sim M$$

2. If $L \preceq M$ and $M \preceq N$ then $L \preceq N$
3. If $L \preceq M \preceq N$then there exists $p \in [0,1]$ such that

$$pL + (1-p)N \sim M$$

4. If $L \prec M$, then for any N and $p \in (0,1]$, one has

$$pL + (1-p)N \prec pM + (1-p)N.$$

The last axiom is called 'independence of irrelevant alternatives'.

An agent whose preferences satisfy the VNM axioms is called 'VNM-rational'. I will leave further discussion of 'what is rationality' to the philosophers, but someone whose

preferences don't satisfy these axioms probably shouldn't be allowed near a race track (or the stock market).

Theorem 1 (Von Neumann and Morgenstern 1945). For any VNM-rational agent (i.e. satisfying 1–4), there exists a function u assigning to each outcome A a real number $u(A)$ such that for any two lotteries,

$$L \prec M \text{ iff } \mathrm{E}(u(L)) < \mathrm{E}(u(M)).$$

Conversely, any agent acting to maximize the expectation of a function u will obey axioms 1–4.

Since

$$\mathrm{E}u(p_1A_1 + \ldots + p_rA_n) = p_1u(A_1) + \cdots + p_nu(A_n).$$

it follows that u is uniquely determined (up to adding a constant and multiplying by a positive scalar) by preferences between *simple lotteries*, i.e. lotteries of the form $pA + (1-p)B$ having only two outcomes.

We now illustrate the intuition behind, and proper use of, utility functions in practical situations requiring decision making under risk. We do this by means of a humorous story of adventure on the high seas. The year is 1776 and you own goods located abroad, worth the equivalent of one standard-size gold bar. These goods cannot increase your wealth until they are shipped back to you via sailing vessels on the high seas, but it's a perilous journey; the probability that a ship is lost at sea is 1/2.

You were planning to have the entire load sent on one ship. Captain Cook advises you that this is unwise and generously offers to split the load in half and send each half on a separate ship at no extra cost. Should you accept Cook's offer?

$$\text{one ship} : \mathbb{E}[w_T] = \frac{1}{2} \times 1 = 0.5 \text{ gold bar}$$

$$\text{two ships} : \mathbb{E}[w_T] = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} = 0.5 \text{ gold bar}$$

You are about to advise Captain Cook that, due to extremely clever use of probability theory, you have proven that it doesn't matter – he can simply use one ship. Just then, Professor Daniel Bernoulli arrives and advises you to instead calculate $\mathrm{E}[u(w_T)]$ where $u(w) = 1 - e^{-w}$, leading to:

$$\text{one ship} : \mathbb{E}[1 - e^{-w_T}] = \frac{1}{2} \times (1 - e^{-1}) \approx 0.32$$

$$\text{two ships} : \mathbb{E}[1 - e^{-w_T}] = \frac{1}{4}(1 - e^{-0}) + \frac{1}{2}(1 - e^{-1/2}) + \frac{1}{4}(1 - e^{-1}) \approx 0.35$$

Using Bernoulli's method, it seems that two ships are preferred, although the reason for the method's efficacy is perhaps still obscure. Bernoulli asks whether you bothered to consider the *risk* when you compared the two scenarios. You reply angrily that

you prefer to act first and consider the risks later. But to make Bernoulli happy, you calculate:

$$\text{one ship}: \mathbb{V}[w_T] = \frac{1}{2}(0-0.5)^2 + \frac{1}{2}(1-0.5)^2$$
$$= 0.25$$
$$\text{two ships}: \mathbb{V}[w_T] = \frac{1}{4}(0-0.5)^2 + \frac{1}{2}(0.5-0.5)^2 + \frac{1}{4}(1-0.5)^2$$
$$= 0.125$$

Bernoulli says that if

$$u(w) = \frac{1-\exp(-\kappa w)}{\kappa}$$

where $\kappa > 0$ is any positive scalar, then supposing w_T is normal,

$$\mathbb{E}[u(w_T)] = u\left(\mathbb{E}[w_T] - \frac{\kappa}{2}\mathbb{V}[w_T]\right) \tag{12.11}$$

This implies that maximizing $\mathrm{E}[u(w_T)]$ is equivalent to maximizing

$$\mathbb{E}[w_T] - \frac{\kappa}{2}\mathbb{V}[w_T] \tag{12.12}$$

since u is monotone. It turns out this is true for many fat-tailed distributions as well, as we show in the next section.

12.4 MEAN-VARIANCE EQUIVALENCE

In the previous section we recalled the well-known result that for an exponential utility function and for normally distributed wealth increments, one may dispense with maximizing $\mathrm{E}[u(w_T)]$ and equivalently solve the mathematically simpler problem of maximizing $\mathrm{E}[w_T] - (\kappa/2)\mathrm{V}[w_T]$. Since this is actually the problem our reinforcement learning systems are going to solve, naturally we'd like to know the class of problems to which it applies. It turns out that neither of the conditions of normality or exponential utility is necessary; both can be relaxed very substantially.

Definition 2 A utility function $u : \mathrm{R} \to \mathrm{R}$ is called *standard* if it is increasing, concave and continuously differentiable.

The properties which define a 'standard' utility function make economic sense. Even great philanthropists have increasing utility of wealth in their investment portfolio – they would prefer to be able to do more to end hunger, disease, etc. Hence a quadratic function that is not linear can never be a standard utility function. A strictly concave quadratic must go up and come back down, as if after some point, more wealth is somehow worse. In particular (12.12) is *not* a utility function.

Concavity corresponds to risk aversion. Finally, if the utility function is not continuously differentiable, it implies that there is a certain particular level of wealth for which one penny above that is very different than one penny below.

Definition 3 Let 'denote a lottery, and let w' denote the (random) final wealth associated to lottery'. For two scalars $m \in \mathrm{R}$ and $s > 0$, let $L(\mu,\omega)$ denote the space of lotteries' under which $\mathrm{E}[w_\prime] = \mu$ and $\mathrm{V}[w_\prime] = \omega^2$. We say *expected utility is a function of mean and variance* if $\mathrm{E}[u(w_\prime)]$ is the same for all '$\in L(\mu,\omega)$. This means that the function $\hat{U}$ defined by

$$\hat{U}(\mu,\omega) := \{\mathrm{E}[u(w')] :' \in L(\mu,\omega)\}$$

is single-valued; the right-hand side is always a single number.

Let $\mathbf{r} \in \mathrm{R}^n$ denote the return over the interval $[t,t+1]$. Hence $\mathbf{r} \in \mathrm{R}^n$ is an n-dimensional vector whose i-th component is

$$r_i = p_i(t+1)/p_i(t) - 1$$

where $p_i(t)$ is the i-th asset's price at time t (adjusted for splits or capital actions if necessary).

Let $\mathbf{h} \in \mathrm{R}^n$ denote the portfolio holdings, measured in dollars or an appropriate numeraire currency, at some time t in the future. Let $\mathbf{h}_0$ denote the current portfolio. Hence the (one-period) wealth random variable is

$$\widetilde{w} = \mathrm{h}'\mathrm{r}$$

and the expected-utility maximizer chooses the optimal portfolio $\mathbf{h}^*$ defined by

$$\mathbf{h}^* := \mathrm{argmax}\mathrm{E}[u(w^{\sim})] \tag{12.13}$$

Definition 4 The underlying asset return distribution, $p(\mathbf{r})$, is said to be *meanvariance equivalent* if first and second moments of the distribution exist, and for any standard utility function u, there exists some constant $\kappa > 0$ (where κ depends on u) such that

$$\mathbf{h}^* = \mathrm{argmax}\{\mathrm{E}[\widetilde{w}] - (\kappa/2)\mathrm{V}[\widetilde{w}]\} \tag{12.14}$$

where $\mathbf{h}^* = \mathrm{argmax}\mathrm{E}[u(\widetilde{w})]$ as defined by (12.13).

The multivariate Cauchy distribution is elliptical, but its moments of orders 1 and higher are all infinite/undefined. Therefore, it is not mean-variance equivalent because the requisite means and variances would be undefined.

Which distributions, then, are mean-variance equivalent? We showed previously that the normal distribution is; this is easy. Many distributions, including heavy-tailed distributions such as the multivariate Student-t, are also mean-variance equivalent.

Assuming all lotteries correspond to holding portfolios of risky assets, then Definition 3, like Definition 4, is a property of the asset return distribution $p(\mathbf{r})$; some distributions have this property and some do not.

If Definition 3 does *not* hold for a given distribution, then there isn't much hope for mean-variance equivalence to hold either. Intuitively, if Definition 3 does *not* hold then $\mathrm{E}[u(w_\prime)]$ must depend on something apart from $\mathrm{E}[w_\prime]$ and $\mathrm{V}[w']$ so it should be easy to construct a counterexample where the right-hand side (12.14) is suboptimal because of this 'extra term'.

Definition 5 An *indifference curve* is a level curve of the surface $\widehat{U}$, or equivalently a set of the form $\widehat{U}^{-1}(c)$

The intuition behind the terminology of Definition 5 is that the investor is indifferent among the outcomes described by the various points on the curve.

Tobin (1958) assumed that expected utility is a function of mean and variance and showed mean-variance equivalence as a consequence. Unfortunately, Tobin's proof was flawed – it contained a derivation which is valid only for elliptical distributions. The flaw in Tobin's proof, and a counterexample, was pointed out by Feldstein (1969). After presenting a correct proof, we will discuss the flaw.

Recall that for a scalar-valued random variable X, the *characteristic function* is defined by

$$\phi_X(t) = \mathbb{E}[e^{itX}],$$

If the variable has a density, then the characteristic function is the Fourier transform of the density. The characteristic function of a real-valued random variable always exists, since it is the integral of a bounded continuous function over a finite measure space.

Generally speaking, characteristic functions are especially useful when analyzing moments of random variables and linear combinations of random variables. Characteristic functions have been used to provide especially elegant proofs of some of the key results in probability theory, such as the central limit theorem.

If a random variable X has moments up to order k, then the characteristic function ϕ_X is k times continuously differentiable on R. In this case

$$\mathbb{E}[X^k] = (-i)^k \phi_X^{(k)}(0).$$

If φ_X has a k-th derivative at zero, then X has all moments up to k if k is even, but only up to $k-1$ if k is odd, and

$$\phi_X^{(k)}(0) = i^k \mathbb{E}[X^k]$$

If $X_1, \ldots, X_n$ are independent random variables, then

$$\varphi_{X1} + \ldots + X_n(t) = \varphi_{X1}(t) \cdots \phi_{Xn}(t).$$

Definition 6 An R^n-valued random variable $\mathbf{x}$ is said to be *elliptical* if its characteristic function, defined by $\varphi(\mathbf{t}) = \mathrm{E}[\exp(i\mathbf{t}^0\mathbf{x})]$ takes the form

$$\varphi(\mathbf{t}) = \exp(i\mathbf{t}^0\mu)\psi(\mathbf{t}^0\Omega\mathbf{t}) \tag{12.15}$$

where $\mu \in \mathrm{R}^n$ is the vector of medians, and Ω is a matrix, assumed to be positive definite, known as the *dispersion matrix*. The function ψ does not depend on n.

We denote the distribution with characteristic function (12.15) by $\mathrm{E}_\psi(\mu,\Omega)$.

The name 'elliptical' arose because the isoprobability contours are ellipsoidal. If variances exist, then the covariance matrix is proportional to Ω, and if means exist, μ is also the vector of means.

Equation (12.15) does not imply that the random vector $\mathbf{x}$ has a density, but if it does, then the density must be of the form

$$f_n(\mathrm{x}) = |\Omega|^{-1/2} g_n[(\mathrm{x}-\mu)'\Omega^{-1}(\mathrm{x}-\mu)] \tag{12.16}$$

Equation (12.16) is sometimes used as the definition of elliptical distributions, when existence of a density is assumed. In particular, (12.16) shows that if $n = 1$, then the

$$\sqrt{-}$$

transformed variable $z = (x-\mu)/\sqrt{\Omega}$ satisfies $z \sim \mathrm{E}_\psi(0,1)$.

The multivariate normal is the most well-known elliptical family; for the normal, one has $g_n(s) = c_n \exp(-s/2)$ (where c_n is a normalization constant) and $\psi(T) = \exp(-T/2)$. Note that g_n depends on n while ψ does not. The elliptical class also includes many non-normal distributions, including examples which display heavy tails and are therefore better suited to modelling asset returns. For example, the multivariate Student-t distribution with ν degrees of freedom has density of the form (12.16) with

$$gn(s) \propto (\nu+s) - (n+\nu)/2 \tag{12.17}$$

and for $\nu = 1$ one recovers the multivariate Cauchy.

One could choose $g_n(s)$ to be identically zero for sufficiently large s, which would make the distribution of asset returns bounded above and below. Thus, one criticism of the CAPM – that it requires assets to have unlimited liability – is not a valid criticism.

Let $\mathbf{v} = \mathbf{T}\mathbf{x}$ denote a fixed (non-stochastic) linear transformation of the random vector $\mathbf{x}$. It is of interest to relate the characteristic function of $\mathbf{v}$ to that of $\mathbf{x}$.

$$\begin{aligned}\phi_{\mathrm{v}}(\mathrm{t}) &= \mathbb{E}[e^{i\mathrm{t}'\mathrm{v}}] = \mathbb{E}[e^{i\mathrm{t}'\mathrm{T}\mathrm{x}}] = \phi_{\mathrm{x}}(\mathrm{T}'t) = e^{i\mathrm{t}'\mathrm{T}\mu}\psi(\mathrm{t}'T\Omega\mathrm{T}'\mathrm{t}) \\ &= e^{i\mathrm{t}'\mathrm{m}}\psi(\mathrm{t}'\Delta t)\end{aligned} \tag{12.18}$$

where for convenience we define $\mathbf{m} = \mathbf{T}\mu$ and $\Delta = \mathbf{T}\Omega\mathbf{T}^0$.

Even with the same function ψ, the functions f_n,g_n appearing in the density (12.16) can have rather different shapes for different n (=the dimension of R^n), as we see in (12.17). However, the function ψ does not depend on n. For this reason, one sometimes speaks of an elliptical 'family' which is identified with a single function ψ, but possibly different values of μ,Ω and a family of functions g_n determining the densities – one such function for each dimension of Euclidean space. Marginalization of an elliptical family results in a new elliptical of the same family (i.e. the same ψ-function).

Theorem 2 If the distribution of $\mathbf{r}$ is elliptical, and if u is a standard utility function, then expected utility is a function of mean and variance, and moreover

$$\partial_\mu \widehat{U}(\mu,\omega) \geq 0 \text{ and } \partial_\omega \widehat{U}(\mu,\omega) \leq 0 \tag{12.19}$$

Proof. For the duration of this proof, fix a portfolio with holdings vector $\mathbf{h} \in \mathrm{R}^n$ and let $x = \mathbf{h}^0\mathbf{r}$ denote the wealth increment. Let $\mu = \mathbf{h}^0\mathrm{E}[\mathbf{r}]$ and $\omega^2 = \mathbf{h}^0\Omega\mathbf{h}$ denote moments of x. Applying the marginalization property (12.18) with the $1 \times n$ matrix $\mathbf{T} = \mathbf{h}^0$ yields

$$\varphi_x(t) = e^{it\mu}\psi(t^2\omega^2).$$

The k-th central moment of x will be

$$i^{-k}\frac{d^k}{dt^k}\psi(t^2\omega^2)\bigg|_{t=0}$$

From this it is clear that all odd moments will be zero and the $2k$-th moment will be proportional to ω^{2k}. Therefore the full distribution of x is completely determined by μ,ω, so expected utility is a function of μ,ω.

We now prove the inequalities (12.19). Write

$$\widehat{U}(\mu,\omega) = \mathbb{E}[u(x)] = \int_{-\infty}^{\infty} u(x)f(x)dx. \tag{12.20}$$

Note that the integral is over a one-dimensional variable. Using the special case of Eq. (12.16) with $n = 1$, we have

$$f_1(x) = \omega^{-1}g_1[(x-\mu)^2/\omega^2]. \tag{12.21}$$

Using (12.21) to update (12.20), we have

$$\widehat{U}(\mu,\omega) = \mathbb{E}[u(x)] = \int_{-\infty}^{\infty} (x)\omega^{-1}g_1[(x-\mu)^2/\omega^2]dx.$$

Now make the change of variables $z = (x-\mu)/\omega$ and $dx = \omega\, dz$, which yields

$$\widehat{U}(\mu,\omega) = \int_{-\infty}^{\infty} u(\mu+\omega z)g_1(z^2)dz.$$

The desired property $\partial_\mu \widehat{U}(\mu,\omega) \geq 0$ then follows immediately from the condition from Definition 2 that u is increasing.

The case for $\partial_\omega \widehat{U}$ goes as follows:

$$\begin{aligned}
\partial_\omega \widehat{U}(\mu,\omega) &= \int_{-\infty}^{\infty} \tau s'(\mu+\omega z)zg_1(z^2)dz \\
&= \left[\int_{-\infty}^{0} + \int_{0}^{\infty}\right] u'(\mu+\omega z)zg_1(z^2)dz \\
&= -\int_{0}^{\infty} u'(\mu-\omega z)zg_1(z^2)dz + \int_{0}^{\infty} u'(\mu+\omega z)zg_1(z^2)dz \\
&= \int_{0}^{\infty} zg_1(z^2)[u'(\mu+\omega z) - u'(\mu-\omega z)]dz
\end{aligned}$$

A differentiable function is concave on an interval if and only if its derivative is monotonically decreasing on that interval, hence

$$u^0(\mu+\omega z) - u^0(\mu-\omega z) < 0$$

while $g_1(z^2) > 0$ since it is a probability density function. Hence on the domain of integration, the integrand of $\int_0^\infty z g_1(z^2)[u'(\mu + \omega z) - u'(\mu - \omega z)]dz$ is non-positive and hence $\partial_\omega \hat{U}(\mu, \omega) \leq 0$, completing the proof of Theorem 2.

Recall Definition 5 above of indifference curves. Imagine the indifference curves written in the σ,μ plane with σ on the horizontal axis. If there are two branches of the curve, take only the upper one. Under the conditions of Theorem 2, one can make two statements about the indifference curves:

- $d\mu/d\sigma > 0$ or an investor is indifferent about two portfolios with different variances only if the portfolio with greater σ also has greater μ,
- $d^2\mu/d\sigma^2 > 0$, or the rate at which an individual must be compensated for accepting greater σ (this rate is $d\mu/d\sigma$) increases as σ increases

These two properties say that the indifference curves are convex.

In case you are wondering how one might calculate $d\mu/d\sigma$ along an indifference curve, we may assume that the indifference curve is parameterized by

$$\lambda \to (\mu(\lambda), \sigma(\lambda))$$

and differentiate both sides of

$$\mathrm{E}[u(x)] = u(\mu + \sigma z) g_1[z^2] dz.$$

with respect to λ. By assumption, the left side is constant (has zero derivative) on an indifference curve. Hence

$$0 = \int u'(\mu + \sigma z)(\mu'(\lambda) + z\sigma'(\lambda)) g_1(z^2) dz$$

$$\frac{d\mu}{d\sigma} = \frac{\mu'(\lambda)}{\sigma'(\lambda)} = -\frac{\int_{\mathbb{R}} z u'(\mu + \sigma z) g_1(z^2) dz}{\int_{\mathbb{R}} u'(\mu + \sigma z) g_1(z^2)\, dz}$$

If $u^0 > 0$ and $u^{00} < 0$ at all points, then the numerator ${}^{\mathrm{R}}\mathrm{R}\ z\ u^0(\mu + \sigma z) g_1(z^2) dz$ is negative, and so $d\mu/d\sigma > 0$.

The proof that $d^2\mu/d\sigma^2 > 0$ is similar (exercise).

What, exactly, fails if the distribution $p(\mathbf{r})$ is not elliptical? The crucial step of this proof assumes that a two-parameter distribution $f(x;\mu,\sigma)$ can be put into 'standard form' $f(z;0,1)$ by a change of variables $z = (x - \mu)/\sigma$. This is not a property of all two-parameter probability distributions; for example, it fails for the lognormal.

One can see by direct calculation that for logarithmic utility, $u(x) = \log x$ and for a log-normal distribution of wealth,

$$f(x; m, s) = \frac{1}{sx\sqrt{2\pi}} \exp(-(\log\, x - m)^2/2s^2)$$

then the indifference curves are not convex. The moments of x are

$$\mu = em + s^2/2, \quad \text{and} \quad \sigma 2 = (em + s^2/2)2(es^2 - 1)$$

and with a little algebra one has

$$\mathbb{E}u = \log\ \mu - \frac{1}{2}\log(\sigma^2/\mu^2 + 1)$$

One may then calculate $d\mu/d\sigma$ and $d^2\mu/d\sigma^2$ along a parametric curve of the form

$$\mathrm{E}u = \text{constant}$$

and see that $d\mu/d\sigma > 0$ everywhere along the curve, but $d^2\mu/d\sigma^2$ changes sign. Hence this example cannot be mean-variance equivalent.

Theorem 2 implies that for a given level of median return, the right kind of investors always dislike dispersion. We henceforth assume, unless otherwise stated, that the first two moments of the distribution exist. In this case (for elliptical distributions), the median is the mean and the dispersion is the variance, and hence the underlying asset return distribution is mean-variance equivalent in the sense of Definition 4. We emphasize that this holds for any smooth, concave utility.

12.5 REWARDS

In some cases the shape of the reward function is not obvious. It's part of the art of formulating the problem and the model. My advice in formulating the reward function is to think very carefully about what defines 'success' for the problem at hand, and in a very complete way. A reinforcement learning agent can learn to maximize only the rewards it knows about. If some part of what defines success is missing from the reward function, then the agent you are training will most likely fall behind in exactly that aspect of success.

12.5.1 The form of the reward function for trading

In finance, as in certain other fields, the problem of reward function is also subtle, but happily this subtle problem has been solved for us by Bernoulli (1954), Von Neumann and Morgenstern (1945), Arrow (1971) and Pratt (1964). The theory of decision making under uncertainty is sufficiently general to encompass very many, if not all, portfolio selection and optimal-trading problems; should you choose to ignore it, you do so at your own peril.

Consider again maximizing (12.12):

$$\text{maximize:} \left\{ \mathbb{E}[\omega_T] - \frac{k}{2}\mathbb{V}[\omega_T] \right\} \tag{12.22}$$

Suppose we could invent some definition of 'reward' R_t so that

$$\mathbb{E}[\omega_T] - \frac{k}{2}\mathbb{V}[\omega_T] \approx \sum_{t=1}^{T} R_t \tag{12.23}$$

Then (12.22) looks like a 'cumulative reward over time' problem.

Reinforcement learning is the search for policies which maximize

$$E[Gt] = E[Rt + 1 + \gamma Rt + 2 + \gamma 2Rt + 3 + \dots]$$

which by (12.23) would then maximize expected utility as long as $\gamma \approx 1$.

Consider the reward function

$$R_t := \delta\omega_t - \frac{k}{2}(\delta\omega_t - \widehat{\mu})^2 \tag{12.24}$$

where $\widehat{\mu}$ is an estimate of a parameter representing the mean wealth increment over one period, $\mu = \mathrm{E}[\delta w_t]$.

$$\frac{1}{T}\sum_{t=1}^{T} R_t = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\delta\omega_t}_{\to \mathbb{E}[\delta\omega_t]} - \underbrace{\frac{k}{2}\frac{1}{T}\sum_{t=1}^{T}(\delta\omega_t - \widehat{\mu})^2}_{\to \mathbb{V}[\delta\omega_t]}$$

Then and for large T, the two terms on the right-hand side approach the sample mean and the sample variance, respectively.

Thus, with this one special choice of the reward function (12.24), if the agent learns to maximize cumulative reward, it should also approximately maximize the mean-variance form of utility.

12.5.2 Accounting for profit and loss

Suppose that trading in a market with N assets occurs at discrete times $t = 0,1,2,\dots,T$. Let $n_t \in \mathrm{Z}^N$ denote the holdings vector in *shares* at time t, so that

$$ht := ntpt \in \mathrm{R}N$$

denotes the vector of holdings in dollars, where p_t denotes the vector of midpoint prices at time t.

Assume for each t, a quantity δn_t shares are traded in the instant just before t and no further trading occurs until the instant before $t+1$. Let

$$v_t = \mathrm{nav}_t + \mathrm{cash}_t \text{ where } \mathrm{nav}_t = n_t \cdot p_t$$

denote the 'portfolio value', which we define to be net asset value in risky assets, plus cash. The *profit and loss* (PL) before commissions and financing over the interval $[t,t+1)$ is given by the change in portfolio value δv_{t+1}.

For example, suppose we purchase $\delta n_t = 100$ shares of stock just before t at a per-share price of $p_t = 100$ dollars. Then nav_t increases by 10 000 while cash_t decreases by 10 000, leaving v_t invariant. Suppose that just before $t+1$, no further trades have occurred and $p_{t+1} = 105$; then $\delta v_{t+1} = 500$, although this PL is said to be *unrealized* until we trade again and move the profit into the cash term, at which point it is *realized*.

Now suppose $p_t = 100$ but due to bid-offer spread, temporary impact or other related frictions our effective purchase price was $\widetilde{p}_t = 101$. Suppose further that we

continue to use the midpoint price p_t to 'mark to market', or compute net asset value. Then, as a result of the trade, nav_t increases by $(\delta n_t)p_t = 10\,000$ while cash_t decreases by 10 100, which means that v_t is decreased by 100 even though the reference price p_t has not changed. This difference is called *slippage*; it shows up as a cost term in the cash part of v_t.

Executing the trade list results in a change in cash balance given by

$$\delta(\text{cash})_t = -\delta n_t \cdot \widetilde{p}_t$$

where $\widetilde{p}_t$ is our effective trade price including slippage. If the components of δn_t were all positive then this would represent payment of a positive amount of cash, whereas if the components of δn_t were negative we receive cash proceeds.

Hence before financing and borrow cost, one has

$$\begin{aligned}
\delta v_t := v_t - v_{t-1} &= \delta(\text{nav})_t + \delta(\text{cash})_t \\
&= n_t \cdot p_t - n_t{-}_1 \cdot p_t{-}_1 - \delta n_t \cdot \widetilde{p}_t && (12.25)\\
&= nt \cdot pt - nt - 1 \cdot pt + nt - 1 \cdot pt - nt - 1 \cdot pt - 1 - \delta nt \cdot \widetilde{p}t && (12.26)\\
&= \delta nt \cdot (pt - \widetilde{p}t) + nt - 1 \cdot (pt - pt - 1) && (12.27)\\
&= \delta n_t \cdot (p_t - \widetilde{p}_t) + h_{t-1} \cdot r_t && (12.28)
\end{aligned}$$

where the asset returns are $r_t = p_t/p_{t-1} - 1$. Let us define the *total cost* c_t inclusive of both slippage and borrow/financing cost as follows:

$$c_t := \text{slip}_t + fin_t, \quad \text{where} \tag{12.29}$$

$$\text{slip}_t := \delta n_t \cdot (\widetilde{p}_t - p_t) \tag{12.30}$$

where fin_t denotes the commissions and financing costs incurred over the period, commissions are proportional to δn_t and financing costs are convex functions of the components of n_t. The component slip_t is called the *slippage cost*. Our conventions are such that $\text{fin}_t > 0$ always, and $\text{slip}_t > 0$ with high probability due to market impact and bid-offer spreads.

12.6 PORTFOLIO VALUE VERSUS WEALTH

Combining (12.29),(12.30) with (12.28) we have finally

$$\delta v_t = h_{t-1} \cdot r_t - c_t \tag{12.31}$$

If we could liquidate the portfolio at the midpoint price vector p_t, then v_t would represent the total wealth at time t associated to the trading strategy under consideration. Due to slippage it is unreasonable to expect that a portfolio can be liquidated at prices p_t, which gives rise to costs of the form (12.30).

Concretely, $v_t = \text{nav}_t + \text{cash}_t$ has a cash portion and a non-cash portion. The cash portion is already in units of wealth, while the non-cash portion $\text{nav}_t = n_t \cdot p_t$ could be converted to cash if a cost were paid; that cost is known as *liquidation slippage*:

$$\text{liqslip}_t := -n_t \cdot (\widetilde{p}_t - p_t)$$

Hence it is the formula for slippage, but with $\delta n_t = -n_t$. Note that liquidation is relevant at most once per episode, meaning the liquidation slippage should be charged at most once, after the final time T.

To summarize, we may identify v_t with the wealth process w_t as long as we are willing to add a single term of the form

$$\text{E}[\text{liqslip}_T] \tag{12.32}$$

to the multi-period objective. If T is large and the strategy is profitable, or if the portfolio is small compared with the typical daily trading volume, then $\text{liqslip}_T \ll v_T$ and (12.32) can be neglected without much influence on the resulting policy. In what follows, for simplicity we identify v_t with total wealth w_t.

12.7 A DETAILED EXAMPLE

Formulating an intelligent behaviour as a reinforcement learning problem begins with identification of the state space S and the action space A. The state variable s_t is a data structure which, simply put, must contain everything the agent needs to make a trading decision, and nothing else. The values of any alpha forecasts or trading signals must be part of the state, because if they aren't, the agent can't use them.

Variables that are good candidates to include in the state:

1. The current position or holding.
2. The values of any signals which are believed to be predictive.
3. The current state of the market microstructure (i.e. the limit order book), so that the agent may decide how best to execute.

In trading problems, the most obvious choice for an action is the number of shares to trade, δn_t, with sell orders corresponding to $\delta n_t < 0$. In some markets there is an advantage to trading round lots, which constrains the possible actions to a coarser lattice. If the agent's interaction with the market microstructure is important, there will typically be more choices to make, and hence a larger action space. For example, the agent could decide which execution algorithm to use, whether to cross the spread or be passive, target participation rate, etc.

We now discuss how the reward is observed during the trading process. Immediately before time t, the agent observes the state p_t and decides an action, which is a trade list δn_t in units of shares. The agent submits this trade list to an execution system and then can do nothing until just before $t+1$.

The agent waits one period and observes the reward

$$R_{t+1} \approx \delta v_{t+1} - \frac{k}{2}(\delta v_{t+1})^2. \tag{12.33}$$

The goal of reinforcement learning in this context is that the agent will learn how to maximize the cumulative reward, i.e. the sum of (12.33) which approximates the mean-variance form $\mathrm{E}[\delta v] - (\kappa/2)\mathrm{V}[\delta v]$.

For this example, assume that there exists a tradable security with a strictly positive price process $p_t > 0$. (This 'security' could itself be a portfolio of other securities, such as an ETF or a hedged relative-value trade.)

Further suppose that there is some 'equilibrium price' p_e such that $x_t = \log(p_t/p_e)$ has dynamics

$$dx_t = -\lambda x_t + \sigma\, \xi_t \tag{12.34}$$

where $\xi_t \sim N(0,1)$ and ξ_t, ξ_s are independent when $t \neq s$. This means that p_t tends to revert to its long-run equilibrium level p_e with mean-reversion rate λ. These assumptions imply something similar to an arbitrage! Positions taken in the appropriate direction while very far from equilibrium have very small probability of loss and extremely asymmetric loss-gain profiles.

For this exercise, the parameters of the dynamics (12.34) were taken to be $\lambda = \log(2)/H$, where $H = 5$ is the half-life, $\sigma = 0.1$ and the equilibrium price is $p_e = 50$.

All realistic trading systems have limits which bound their behaviour. For this example we use a reduced space of actions, in which the trade size δn_t in a single interval is limited to at most K round lots, where a 'round lot' is usually 100 shares (most institutional equity trades are in integer multiples of round lots). Also we assume a maximum position size of M round lots. Consequently, the space of possible trades, and also the action space, is

$$\mathrm{A} = \mathrm{LotSize} \cdot \{-K, -K+1, \ldots, K\}$$

Letting H denote the possible values for the holding n_t, then similarly

$$H = \{-M, -M+1, \ldots, M\}.$$

For the examples below, we take $K = 5$ and $M = 10$.

Another feature of real markets is the *tick size*, defined as a small price increment (such as US$0.01) such that all quoted prices (i.e. all bids and offers) are integer multiples of the tick size. Tick sizes exist in order to balance price priority and time priority. This is convenient for us since we want to construct a discrete model anyway. We use TickSize = 0.1 for our example.

We choose boundaries of the (finite) space of possible prices so that sample paths of the process (12.34) exit the space with vanishingly small probability. With the parameters as above, the probability that the price path ever exits the region [0.1, 100] is small enough that no aspect of the problem depends on these bounds.

Concretely, the space of possible prices is:

$$\mathrm{P} = \mathrm{TickSize} \cdot \{1, 2, \ldots, 1000\} \subset \mathrm{R}+$$

We do not allow the agent, initially, to know anything about the dynamics. Hence, the agent does not know λ, σ, or even that some dynamics of the form (12.34) are valid.

The agent also does not know the trading cost. We charge a spread cost of one tick size for any trade. If the bid-offer spread were equal to two ticks, then this fixed cost would correspond to the slippage incurred by an aggressive fill which crosses the spread to execute. If the spread is only one tick, then our choice is overly conservative. Hence

$$\text{SpreadCost}(\delta n) = \text{TickSize} \cdot |\delta n| \tag{12.35}$$

We also assume that there is permanent price impact which has a linear functional form: each round lot traded is assumed to move the price one tick, hence leading to a dollar cost $|\delta n_t| \times \text{TickSize/LotSize}$ per share traded, for a total dollar cost for all shares

$$\text{ImpactCost}(\delta n) = (\delta n)^2 \times \text{TickSize/LotSize}. \tag{12.36}$$

The total cost is the sum

$$\begin{aligned}&\text{SpreadCost}(\delta n) + \text{ImpactCost}(\delta n)\\ &= \text{TickSize} \cdot |\delta n| + (\delta n)^2 \times \text{TickSize/LotSize}.\end{aligned}$$

Our claim is not that these are the exact cost functions for the world we live in, although the functional form does make some sense.

The state of the environment $s_t = (p_t, n_{t-1})$ will contain the security prices p_t, and the agent's position, in shares, coming into the period: n_{t-1}. Therefore the state space is the Cartesian product $\text{S} = \text{H} \times \text{P}$. The agent then chooses an action

$$a_t = \delta n_t \in A$$

which changes the position to $n_t = n_{t-1} + \delta n_t$ and observes a profit/loss equal to

$$\delta vt = nt(pt + 1 - pt) - ct$$

and a reward

$$R_{t+1} = \delta v_{t+1} - \frac{1}{2} k(\delta v_{t+1})^2$$

as in Eq. (12.33).

We train the Q-learner by repeatedly applying the update procedure involving (12.9). The system has various parameters which control the learning rate, discount rate, risk aversion, etc. For completeness, the parameter values used in the following example were: $k = 10^{-4}$, $\gamma = 0.999$, $\alpha = 0.001$, $\varepsilon = 0.1$. We use $n_{train} = 10^7$ training steps (each 'training step' consists of one action-value update as per (12.9)) and then evaluate the system on 5000 new samples of the stochastic process (see Figure 12.2).

The excellent performance out-of-sample should perhaps be expected; the assumption of an Ornstein-Uhlenbeck process implies a near arbitrage in the system. When the price is too far out of equilibrium, a trade betting that it returns to equilibrium has a very small probability of a loss. With our parameter settings, even after costs this is

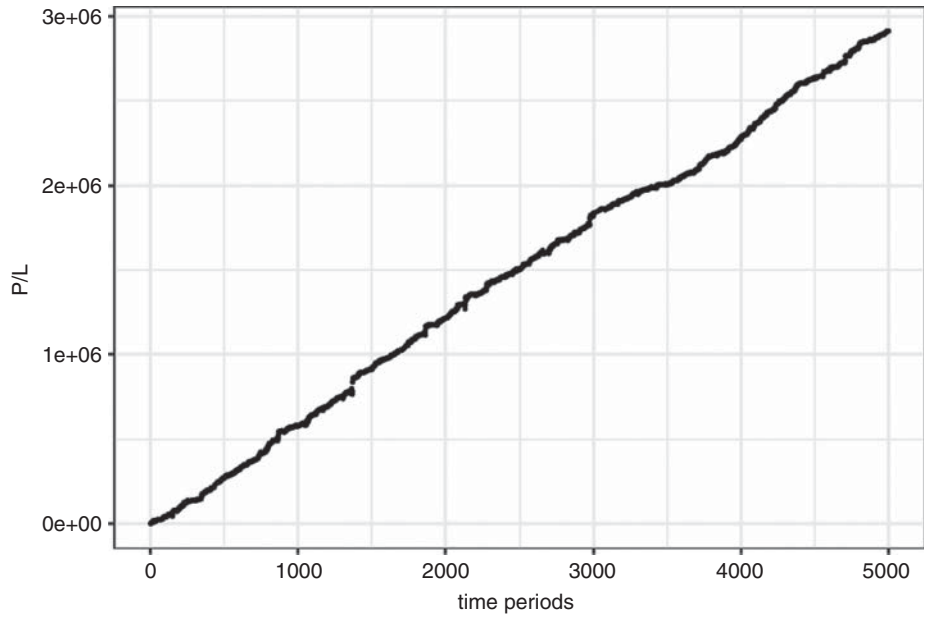

FIGURE 12.2 Cumulative simulated out-of-sample P/L of trained model. Simulated net P/L over 5000 out−of−sample periods.

true. Hence the *existence* of an arbitrage-like trading strategy in this idealized world is not surprising, and perfect mean-reverting processes such as (12.34) need not exist in real markets.

Rather, the surprising point is that the Q-learner does not, at least initially, know that there is mean-reversion in asset prices, nor does it know anything about the cost of trading. At no point does it compute estimates for the parameters λ,σ. It learns to maximize expected utility in a model-free context, i.e. directly from rewards rather than indirectly (using a model).

We have also verified that expected utility maximization achieves a much higher out-of-sample Sharpe ratio than expected-profit maximization. Understanding of this principle dates back at least to 1713 when Bernoulli pointed out that a wealth-maximizing investor behaves nonsensically when faced with gambling based on a martingale (see Bernoulli (1954) for a recent translation).

12.7.1 Simulation-based approaches

A major drawback of the procedure we have presented here is that it requires a large number of training steps (a few million, on the problem we presented). There are, of course, financial datasets with millions of time-steps (e.g. high-frequency data sampled once per second for several years), but in other cases, a different approach is needed. Even in high-frequency examples, one may not wish to use several years' worth of data to train the model.

Fortunately, a simulation-based approach presents an attractive resolution to these issues. We propose a multi-step training procedure:

1. Posit a reasonably parsimonious stochastic process model for asset returns with relatively few parameters.
2. Estimate the parameters of the model from market data, ensuring reasonably small confidence intervals for the parameter estimates.
3. Use the model to simulate a much larger dataset than the real world presents.
4. Train the reinforcement-learning system on the simulated data.

For the model $dx_t = -\lambda x_t + \sigma\, \xi_t$, this amounts to estimating λ,σ from market data, which meets the criteria of a parsimonious model.

The 'holy grail' would be a fully realistic simulator of how the market microstructure will respond to various order-placement strategies. In order to be maximally useful, such a simulator should be able to accurately represent the market impact caused by trading too aggressively.

With these two components – a random-process model of asset returns and a good microstructure simulator – one may generate a training dataset of arbitrarily large size. The learning procedure is then only partially model-free: it requires a model for asset returns but no explicit functional form to model trading costs. The 'trading cost model' in this case would be provided by the market microstructure simulator, which arguably presents a much more detailed picture than trying to distil trading costs down into a single function.

We remark that automatic generation of training data is a key component of AlphaGo Zero (Silver et al. 2017), which was trained primarily by means of self-play – effectively using previous versions of itself as a simulator. Whether or not a simulator is used to train, the search continues for training methods which converge to a desired level of performance in fewer time-steps. In situations where all of the training data is real market data, the number of time-steps is fixed, after all.

12.8 CONCLUSIONS AND FURTHER WORK

In the present chapter, we have seen that reinforcement learning concerns the search for good value functions (and hence, good policies). In almost every subfield of finance, this kind of model of optimal behaviour is fundamental. For examples, most of the classic works on market microstructure theory (Glosten and Milgrom 1985; Copeland and Galai 1983; Kyle 1985) model both the dealer and the informed trader as optimizing their cumulative monetary reward over time. In many cases the cited authors assume the dealer simply trades to maximize expected profit (i.e. is risk-neutral). In reality, no trader is risk-neutral, but if the risk is controlled in some other way (e.g. strict inventory controls) and the risk is very small compared to the premium the dealer earns for their market-making activities, then risk-neutrality may be a good approximation.

Recent approaches to multi-period optimization (Gârleanu and Pedersen 2013; Kolm and Ritter 2015; Benveniste and Ritter 2017; Boyd et al. 2017) all follow value-function approaches based around Bellman optimality.

Option pricing is based on dynamic hedging, which amounts to minimizing variance over the lifetime of the option: variance of a portfolio in which the option is hedged with the replicating portfolio. With transaction costs, one actually needs to solve this multi-period optimization rather than simply looking at the current option Greeks. For some related work, see Halperin (2017).

Therefore, we believe that the search for and use of good value functions is perhaps one of the most fundamental problems in finance, spanning a wide range of fields from microstructure to derivative pricing and hedging. Reinforcement learning, broadly defined, is the study of how to solve these problems on a computer; as such, it is fundamental as well.

An interesting arena for further research takes inspiration from classical physics. Newtonian dynamics represent the greedy policy with respect to an action-value function known as *Hamilton's principal function*. Optimal execution problems for portfolios with large numbers (thousands) of assets are perhaps best treated by viewing them as special cases of Hamiltonian dynamics, as showed by Benveniste and Ritter (2017). The approach in Benveniste and Ritter (2017) can also be viewed as a special case of the framework above; one of the key ideas there is to use a functional method related to gradient descent with regard to the value function. Remarkably, even though it starts with continuous paths, the approach in Benveniste and Ritter (2017) generalizes to treat market microstructure, where the action space is always finite.

Anecdotally, it seems that several of the more sophisticiated algorithmic execution desks at large investment banks are beginning to use reinforcement learning to optimize their decision making on short timescales. This seems very natural; after all, reinforcement learning provides a natural way to deal with the discrete action spaces presented by limit order books which are rich in structure, nuanced and very discrete. The classic work on optimal execution, Almgren and Chriss (1999), does not actually specify how to interact with the order book.

If one considers the science of trading to be split into (1) large-scale portfolio allocation decisions involving thousands of assets and (2) the theory of market microstructure and optimal execution, then both kinds of problems can be unified under the framework of optimal control theory (perhaps stochastic). The main difference is that in problem (2), the discreteness of trading is of primary importance: in a double-auction electronic limit order book, there are only a few price levels at which one would typically transact at any instant (e.g. the bid and the offer, or conceivably nearby quotes) and only a limited number of shares would transact at once. In large-scale portfolio allocation decisions, modelling portfolio holdings as continuous (e.g. vectors in R^n) usually suffices.

Reinforcement learning handles discreteness beautifully. Relatively small, finite action spaces such as those in the games of Go, chess and Atari represent areas where reinforcement learning has achieved superhuman performance. Looking ahead to the next 10 years, we therefore predict that among the two research areas listed above, although reward functions like (12.24) apply equally well in either case, it is in the areas of market microstructure and optimal execution that reinforcement learning will be most useful.

REFERENCES

Almgren, R. and Chriss, N. (1999). Value under liquidation. *Risk* 12 (12): 61–63.

Arrow, K.J. (1971). *Essays in the Theory of Risk-Bearing*. North-Holland, Amsterdam.

Baird, L.C. III and Moore, A.W. (1999). Gradient descent for general reinforcement learning. *Advances in Neural Information Processing Systems* 968–974.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Benveniste, E.J. and Ritter, G. (2017). Optimal microstructure trading with a long-term utility function. https://ssrn.com/abstract=3057570.

Bernoulli, D. (1954). Exposition of a new theory on the measurement of risk. *Econometrica: Journal of the Econometric Society* 22 (1): 23–36.

Bottou, L. (2012). Stochastic gradient descent tricks. In: *Neural Networks: Tricks of the Trade*, 421–436. Springer.

Boyd, S. et al. (2017). Multi-Period Trading via Convex Optimization. *arXiv preprint arXiv:1705.00109*.

Copeland, T.E. and Galai, D. (1983). Information effects on the bid-ask spread. *The Journal of Finance* 38 (5): 1457–1469.

Feldstein, M.S. (1969). Mean-variance analysis in the theory of liquidity preference and portfolio selection. *The Review of Economic Studies* 36 (1): 5–12.

Gârleanu, N. and Pedersen, L.H. (2013). Dynamic trading with predictable returns and transaction costs. *The Journal of Finance* 68 (6): 2309–2340.

Glosten, L.R. and Milgrom, P.R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics* 14 (1): 71–100.

Halperin, I. (2017). QLBS: Q-Learner in the Black-Scholes (–Merton) Worlds. *arXiv preprint arXiv:1712.04609*.

Hasbrouck, J. (2007). *Empirical Market Microstructure*, vol. 250. New York: Oxford University Press.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kolm, P.N. and Ritter, G. (2015). Multiperiod portfolio selection and bayesian dynamic models. *Risk* 28 (3): 50–54.

Kyle, A.S. (1985). Continuous auctions and insider trading. *Econometrica: Journal of the Econometric Society* 53 (6): 1315–1335.

Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518 (7540): 529.

Pratt, J.W. (1964). Risk aversion in the small and in the large. *Econometrica: Journal of the Econometric Society* 32 (1–2): 122–136.

Ritter, G. (2017). Machine learning for trading. *Risk* 30 (10): 84–89. https://ssrn.com/abstract=3015609.

Robbins, H. and Siegmund, D. (1985). A convergence theorem for non negative almost supermartingales and some applications. In: *Herbert Robbins Selected Papers*, 111–135. Springer.

Silver, D. et al. (2017). Mastering the game of go without human knowledge. *Nature* 550 (7676): 354–359.

Sutton, R.S. and Barto, A.G. (1998). *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction*. Second edition, in progress. Cambridge: MIT Press http://incompleteideas.net/book/bookdraft2018jan1.pdf.

Tobin, J. (1958). Liquidity preference as behavior towards risk. *The Review of Economic Studies* 25 (2): 65–86.

Von Neumann, J. and Morgenstern, O. (1945). *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press.

Watkins, C.J.C.H. (1989). Q-learning. *PhD Thesis*.