Università
Ca'Foscari
Venezia

Corso di Laurea magistrale

in Economia e Finanza

Tesi di Laurea

QL e SARSA per il trading finanziario:

nuove funzioni di reward

Relatore
Ch. Prof. Marco Corazza

Correlatrice
Ch.ma Prof.ssa Marta Cardin

Laureanda
Ludovica Moro
Matricola 851542

Anno Accademico
2019 / 2020

Indice

Introduzione

Una vecchia battuta racconta di un economista che passeggia per strada con un amico, quando si imbattono in una banconota da 100$ stesa a terra e mentre l'amico allunga la mano per prenderla, l'economista dice: "Non preoccuparti, se fosse una banconota da 100$, qualcuno l'avrebbe già raccolta". Questo esempio di logica economica è una rappresentazione dell'ipotesi dei mercati efficienti (EMH), una delle teorie più affermate e, allo stesso tempo, contestate in tutte le scienze sociali. Secondo questa teoria, per usare le parole di Fama (1970), "*Prices fully reflect all available information*", cioè i prezzi riflettono perfettamente le informazioni disponibili sul mercato. Questa teoria sostiene che gli agenti economici che operano nel mercato sono completamente razionali, cioè sono in grado di variare istantaneamente e appropriatamente i prezzi delle attività finanziarie semplicemente sulla base delle informazioni recenti, attraverso la legge della domanda e dell'offerta. L'unica fonte di variazioni imprevedibili dei prezzi delle attività finanziarie tra due istanti consecutivi è rappresentata dall'arrivo di nuove informazioni inattese. Su questa teoria, anche dopo aver riscontrato gran successo, sono state compiute ricerche per diversi decenni, che non hanno permesso di ottenere il consenso tra gli economisti sul fatto che i mercati finanziari siano effettivamente efficienti. Come suggerisce il buon senso e le evidenze della finanza comportamentale, gli esseri umani spesso non sono completamente razionali quando prendono decisioni, specialmente in condizioni di incertezza. Dagli anni Settanta gli economisti sperimentali hanno documentato diverse deviazioni dei comportamenti degli investitori reali da quelli prescritti dall'EMH. La principale implicazione che deriva da questi scostamenti dall'efficienza di EMH consiste nel fatto che i mercati finanziari non sono così raramente inefficienti ma offrono, più o meno spesso, la possibilità di una negoziazione redditizia.

Alla luce di queste evidenze, nel 2004 Andrew W. Lo propone una teoria alternativa all'ipotesi dei mercati efficienti, chiamata ipotesi dei mercati adattivi (AMH), che si basa su un approccio evolutivo applicato alle interazioni economiche, nonché su alcuni risultati dell'economia comportamentale (cioè gli effetti di fattori psicologici, cognitivi, emotivi, ecc. sulle decisioni degli individui). Il punto innovativo di questa teoria risiede nella sua ottica evolutiva, secondo la quale gli individui prendono delle scelte basate sull'esperienza passata e sulla loro migliore ipotesi su ciò che potrebbe essere ottimale (almeno ai loro occhi), imparando dagli effetti che generano le loro azioni. Quest'ottica risiede alla base del processo evolutivo e descrive dei meccanismi che in natura avvengono istintivamente; si pensi al banale esempio di un bambino che tocca il fuoco e, scottandosi, impara che ci si brucia. Secondo l'AMH un mercato finanziario può essere visto come un ambiente evolutivo in cui diverse "specie" parzialmente razionali ma comunque intelligenti (ad esempio, hedge funds, market maker, fondi pensione, investitori al dettaglio, speculatori e così via) interagiscono tra loro al fine di raggiungere un livello soddisfacente – non necessariamente ottimale – di una certa misura della redditività. Poichè queste specie sono solo in parte razionali, le loro azioni non sono né istantanee né perfettamente appropriate, e questo generalmente non implica l'efficienza del mercato finanziario. Per questo motivo, l'AMH afferma che da una prospettiva evolutiva, l'esistenza di mercati finanziari inefficienti implica la presenza di opportunità di profitto che scompaiono man mano che vengono sfruttate, ma nuove opportunità vengono costantemente create fintanto che alcune specie si estinguono, altre nascono oppure quando cambiano le istituzioni e le condizioni commerciali (Lo, 2004).

L'elaborato si propone di investigare la valenza della teoria dei mercati adattivi, cioè vuole studiare dei meccanismi in grado di costruire una strategia ottimale per un problema decisionale sequenziale interagendo direttamente con l'ambiente (i mercati), così da cogliere le opportunità di profitto in tempo reale e di apprendere le dinamiche che caratterizzano nel tempo i mercati finanziari.

Negli ultimi anni l'impatto dei sistemi di trading automatizzato sui mercati finanziari è in crescita costante e le negoziazioni generate da un algoritmo

rappresentano ora la maggior parte degli ordini che arrivano in borsa. Uno strumento adatto al nostro scopo si trova nel campo dell'Intelligenza Artificiale, il quale ha sviluppato metodi e strumenti innovativi per la risoluzione dei problemi finanziari e di processo decisionale. Tra questi, si trova un sistema automatizzato di trading finanziario costruito con un approccio auto-adattivo di apprendimento automatico noto come *Reinforcement Learning* (abbraviato con RL) o apprendimento per rinforzo. Nella parte sperimentale si applicano i più noti algoritmi di RL conosciuti con i nomi di Q-Learning (QL) e SARSA. Di questi si propongono due possibili alternative rispetto alla struttura più classica degli algoritmi, che utilizzano lo Sharpe ratio come misura di performance: dato che in letteratura viene tanto applicato quanto descritto come un indicatore soggetto a diversi limiti, si vuole indagare se l'indice di Sortino e di Burke possano essere considerate delle valide alternative.

Questo elaborato è strutturato come segue. Il primo capitolo presenta la teoria dei mercati efficienti (EMH) e, dopo aver accennato ad alcune delle critiche della teoria, l'alternativa ipotesi dei mercati adattivi (AMH). Al secondo capitolo si introduce il Reinforcement Learning in modo qualitativo, mentre al terzo capitolo si approfondiscono gli aspetti quantitativi, attraverso la descrizione del framework matematico di riferimento, cioè i processi di Markov. Il quarto capitolo precisa i dettagli degli elementi utilizzati nella prova pratica, della quale vengono esposti e discussi i risultati al quinto capitolo. Infine, le conclusioni dell'elaborato sono tratte al sesto capitolo.

Capitolo 1

Teoria dei mercati efficienti e teoria dei mercati adattivi: analisi nei sistemi di trading

La maggior parte degli economisti finanziari sono sostenitori dell'ipotesi dei mercati efficienti (dall'inglese *Efficient Market Hypothesis*, abbreviata con EMH, di Samuelson 1965), in cui i prezzi sono determinati dal trading competitivo di molti investitori e tale trading elimina qualsiasi vantaggio informativo che potrebbe esistere tra tutti i membri della comunità di investimento. Secondo l'EMH, il risultato è un mercato in cui i prezzi riflettono pienamente tutte le informazioni disponibili e pertanto non sono prevedibili. Per usare le parole di Fama (1965): "*In an efficient market, on the average, competition will cause the full effects of new information on intrinsic values to be reflected instantaneously in actual prices*".

I critici dell'ipotesi dei mercati efficienti sostengono che gli investitori sono spesso, se non sempre, irrazionali, che mostrano pregiudizi prevedibili e finanziariamente rovinosi come la fiducia eccessiva (Barber e Odean, 2001; Gervais e Odean, 2001; Fischoff e Slovic, 1980), la reazione eccessiva (De Bond e Thaler, 1986), l´avversione alla perdita (Odean, 1998; Shefrin e Statman, 1985; Kahneman e Tversky, 1979), l´*herding* (Huberman e Regev, 2001), la contabilità psicologica (Tversky e Kahneman, 1981), la calibrazione errata delle probabilità (Lichtenstein, Fischoff, e Phillips, 1982), e il rimpianto (Clarke, Krase, e Statman, 1994; Bell, 1982). Le fonti di queste irrazionalità sono spesso attribuite a fattori psicologici: paura, avidità e altre risposte emotive alle fluttuazioni dei prezzi e ai drammatici cambiamenti nella ricchezza di un investitore. Un numero crescente di

economisti, psicologi e professionisti del settore finanziario ha iniziato a usare i termini “economia comportamentale” e “finanza comportamentale” (Shefrin, 2001). Tuttavia, ricerche nelle scienze cognitive e nell'economia finanziaria suggeriscono un legame importante tra la razionalità nel processo decisionale e le emozioni (Loewenstein, 2000; Peters e Slovic 2000; Lo, 1999; Elster, 1998; Damasio, 1994; Grossberg e Gutowski, 1987), sottintendendo che le due nozioni non sono antitetiche ma, di fatto, complementari.

In questo capitolo si presenta prima la teoria dei mercati efficienti, alcune critiche ed evidenze studiate negli ultimi decenni che hanno portato allo sviluppo della finanza comportamentale, infine conclude con la presentazione della teoria dei mercati adattivi di Andrew W. Lo.

1.1 La teoria dei mercati efficienti (EMH) e lo sviluppo della finanza comportamentale

L'ipotesi di mercato efficiente è stata ampiamente accettata dagli economisti finanziari accademici. Uno di questi fu Eugene Fama (1970) con un suo influente articolo intitolato "*Efficient Capital Markets*". Questa teoria sostiene che i mercati dei titoli sono estremamente efficienti nel riflettere le informazioni sui singoli titoli e sul mercato azionario nel suo insieme. L'opinione accolta sostiene che quando si presentano le informazioni, le notizie si diffondono molto rapidamente e vengono incorporate senza indugio nei prezzi dei titoli. Pertanto, né l'analisi tecnica, che è lo studio dei prezzi azionari passati nel tentativo di prevedere i prezzi futuri, né l'analisi fondamentale, che è l'analisi di informazioni finanziarie come gli utili delle società e i valori delle attività per aiutare gli investitori a selezionare titoli sottovalutati, consentirebbe a un investitore di conseguire rendimenti superiori a quelli che si potrebbero ottenere detenendo un portafoglio selezionato casualmente di singoli titoli, almeno non con un rischio comparabile.

La storia dell'ipotesi di mercato efficiente (EMH) può essere suddivisa in tre fasi: una prima costruzione della teoria negli anni ′60, l'istituzione di una conferma empirica che rese consensuale la teoria negli anni ′70 e, infine, l'aumento degli studi empirici che sfidano la teoria dagli anni ′80. Quest'ultimo

passo porta alla produzione di approcci alternativi come la finanza comportamentale (Thaler 1999; Shiller 2003) o, più recentemente, l'ipotesi del mercato adattivo (Lo, 2004).

Secondo la letteratura, la paternità di EMH è attribuita alle opere di Eugene Fama (1965a[1]; 1965b[2]) e Paul Samuelson (1965a)[3] che spiegano il carattere casuale dei prezzi come conseguenza di comportamenti razionali. In realtà, i due autori spiegano la casualità della variazione dei prezzi, eppure producono una spiegazione molto diversa di questo fenomeno. Secondo Fama, l'EMH è un mercato competitivo composto da agenti razionali, in cui il prezzo converge ad un cosiddetto "valore fondamentale", spiegando il carattere casuale del prezzo; secondo Samuelson, la casualità della variazione dei prezzi può essere semplicemente spiegata dalla concorrenza tra agenti razionali senza alcun riguardo per questo valore fondamentale.[4]

L'ipotesi di mercato efficiente è associata all'idea di *random walk*, che è un termine usato nella letteratura finanziaria per caratterizzare una serie di prezzi in cui tutte le variazioni rappresentano scostamenti casuali dai prezzi precedenti. La logica del *random walk* è che se il flusso di informazioni non viene ostacolato e le informazioni si riflettono immediatamente nei prezzi delle azioni, allora la variazione dei prezzi di domani rifletterà solo le notizie di domani e sarà indipendente dalle variazioni dei prezzi di oggi. Ma le notizie sono per definizione imprevedibili e quindi le conseguenti variazioni di prezzo devono essere imprevedibili e casuali. Di conseguenza, i prezzi riflettono appieno tutte le informazioni note e anche gli investitori non informati che acquistano un portafoglio otterranno un tasso di rendimento generoso come quello raggiunto dagli esperti.

---

[1] Fama, E. (1965a). The Behavior of Stock-Market Prices. *The Journal of Business*, 38(1), 34-105.
[2] Fama, E. (1965b). Random Walks in Stock Market Prices. Selected Papers of the Graduate School of Business, University of Chicago, Reprinted in the *Financial Analysts Journal*, September-October 1965; *The Analysts Journal*, London, 1966; *The Institutional Investor*, October 1968, 55-59.
[3] Samuelson, Paul A. (1965). Proof That Properly Anticipated Prices Fluctuate Randomly. *Industrial Management Review*, 6(2), 41-49.
[4] Ai fini di questo elaborato, le differenze tra le teorie di Fama e Samuelson non vengono approfondite. Si veda: Delcey, T. (2019). Samuelson vs Fama on the Efficient Market Hypothesis: The Point of View of Expertise. *Œconomia* , 9-1.

Gli investitori e i ricercatori hanno contestato l'ipotesi del mercato efficiente sia empiricamente che teoricamente. Gli economisti comportamentali[5] attribuiscono le imperfezioni nei mercati finanziari a una combinazione di pregiudizi cognitivi come eccessiva fiducia, eccessiva reazione, propensione rappresentativa, distorsione delle informazioni e vari altri errori umani nel ragionamento e nell'elaborazione delle informazioni. Questi sono stati studiati da psicologi come Daniel Kahneman, Amos Tversky e Paul Slovic ed economisti come Richard Thaler[6]. L'interesse per la psicologia e il comportamento degli investitori è aumentato e ha generato un gran numero di studi e opere, che hanno stabilito negli anni la finanza comportamentale come paradigma dominante nella finanza. A parte l'inquadramento e le anomalie del mercato, una premessa importante in finanza comportamentale è l'euristica[7] che consiste in modelli riguardanti il comportamento delle persone. L'euristica si riferisce all'acquisizione di conoscenza o di un risultato desiderabile impiegando congetture intelligenti anziché formule specificate ed implica semplici tecniche basate sull'esperienza per la risoluzione dei problemi, note come regole empiriche o scorciatoie, che sono state proposte per spiegare come gli investitori prendono decisioni, in particolare durante i periodi in cui, a causa della scarsa informazione, delle complesse circostanze di investimento e dell'instabilità del mercato, è difficile esprimere un giudizio. È consiste in processi che avvengono senza consapevolezza cosciente. Si riportano degli esempi.

Gli autori Odean e Gervais[8] nel 2001 studiano l'effetto dell'*overconfidence* degli investitori, cioè dell'eccessiva confidenza dei traders che cambia a seconda dei propri successi e fallimenti e a seconda del periodo della vita del trader. Questo modello consente di fare previsioni su quando un trader ha maggiori probabilità di essere troppo fiducioso (secondo gli autori, quando è inesperto e di successo) e su come l'eccesso di fiducia cambierà durante la vita di un trader (secondo gli autori aumenta, in media, all'inizio della carriera di un

[5] Un economista comportamentale studia l'economia attraverso una prospettiva psicologica.
[6] Kahneman e Thaler vincitori del Premio Nobel per l'economia, rispettivamente, nel 2002 e 2007.
[7] Euristica: termine (dal greco heurískein, «scoprire, trovare») che, nelle scienze ipotetico-deduttive come la matematica indica l'analisi delle strategie che conducono a risolvere problemi, scoprire proprietà, conseguire risultati da dimostrare poi in modo rigoroso (Treccani).
[8] Gervais S., Odean T., (2001). Learning to Be Overconfident, *The Review of Financial Studies*, 14(1), 1–27.

trader e poi gradualmente diminuisce). Il modello ha anche implicazioni per il cambiamento delle condizioni di mercato. Ad esempio, la maggior parte dei partecipanti al mercato azionario ha posizioni lunghe e beneficia di movimenti al rialzo dei prezzi. Ci si aspetta quindi che l'eccessiva fiducia aggregata sia più elevata dopo i guadagni del mercato e inferiore dopo le perdite del mercato. Poiché una maggiore fiducia eccessiva porta a un maggiore volume degli scambi, ciò suggerisce che il volume degli scambi sarà maggiore dopo gli utili del mercato e inferiore dopo le perdite del mercato. Nello stesso anno Barber e Gervais (2001)[9] studiano il fenomeno dell'eccessiva fiducia tra genere, affermando che in generale gli uomini compiono più transazioni delle donne riducendo di conseguenza i loro rendimenti. Queste differenze sono più pronunciate tra uomini e donne single. Gli autori ritengono inoltre che l'eccessiva fiducia sia la spiegazione per gli alti livelli di contrattazione controproducente nei mercati finanziari.

Uno studio sull'efficienza del mercato compiuto da De Bond e Thaler (1985)[10] scopre che la tendenza delle persone a reagire in modo eccessivo agli eventi di cronaca inaspettati e drammatici influisce sui prezzi delle azioni.

Gli psicologi Kahneman e Tversky nel 1979 pubblicano una critica della teoria dell'utilità attesa come modello del processo decisionale in situazioni di rischio e sviluppa un modello alternativo, chiamato *Prospect Theory*. Con questa pubblicazione[11], gli autori sostengono che le persone sottostimano i risultati di eventi "solamente" probabili rispetto ai risultati che si ottengono con certezza. Questa tendenza contribuisce all'avversione al rischio nelle scelte che comportano guadagni sicuri, alla ricerca del rischio nelle scelte che comportano perdite certe. Sulla base di questo studio, l'autore Odean nel 2002 pubblica un articolo[12] nel quale mette alla prova l'effetto disposizione, cioè la tendenza degli investitori a trattenere investimenti perdenti troppo a lungo e vendere investimenti vincenti

[9]Barber, Brad e Odean, Terrance. (2001). Boys Will Be Boys: Gender, Overconfidence, And Common Stock Investment. *The Quarterly Journal of Economics*. 116. 261-292.
[10] Werner F. M. De Bondt, e Thaler, R. (1985). Does the Stock Market Overreact? *The Journal of Finance*, 40(3), 793-805.
[11] Kahneman, Daniel e Tversky, Amos, (1979). "Prospect Theory: An Analysis of Decision under Risk," *Econometrica*, Econometric Society, 47(2), 263-291.
[12] Odean, T. (1998), Are Investors Reluctant to Realize Their Losses? The Journal of Finance, 53: 1775-1798.

troppo presto. Sullo stesso tema scrivono Shefrin e Statman nel 1985[13]. Ancora, Kahneman e Tversky nel 1981[14] studiano il fenomeno chiamato contabilità mentale (o psicologica) secondo il quale un individuo può valutare due guadagni monetari identici in modo diverso perché sono codificati e valutati attraverso due distinti conti mentali. In particolare, il risparmio su un bene sembra essere più attraente tanto più alto è il suo valore relativo rispetto al costo originale dell'articolo, anche se la spesa totale per tutti gli acquisti effettuati rimane invariata.

Infine, Clarke, Krase, e Statman nel 1994[15] studiano l'effetto del rimpianto sulle performance, notando che gli investitori più avversi al rimpianto si discostano meno dal portafoglio utilizzato come riferimento, dunque subiscono meno rimpianti e allo stesso tempo ricevono un rendimento atteso inferiore rispetto agli investitori che sono invece meno avversi al rimpianto.

Quindi, la finanza comportamentale emerge come un modello che, non solo ha migliorato le teorie finanziarie stagnanti, ma le ha anche confutate, ed è riuscita a sfidare l'attenzione accademica e scientifica per poi essere riconosciuta come un quadro teorico alternativo. Questo approccio accetta le debolezze comportamentali delle persone e afferma che i fallimenti degli investimenti sono una conseguenza naturale dei tratti che caratterizzano il comportamento umano. In questo quadro, la finanza comportamentale tratta gli investitori come individui e sottolinea che emozioni, pregiudizi e illusioni non possono essere razionalizzati; inoltre, sottolinea che le informazioni sono inefficienti. I prezzi delle azioni non sono casuali, sono piuttosto imprevedibili poiché anche la reazione delle persone alle nuove informazioni è imprevedibile. Infine, la finanza comportamentale postula che gli investitori non possono essere esclusi dal proprio passato di investimento in quanto sono esseri umani e, per gli esseri umani, le azioni passate sono una parte vitale della propria storia. In questa prospettiva, i prezzi passati e i

[13] Shefrin, H. e Statman, M. (1985), The Disposition to Sell Winners Too Early and Ride Losers Too Long: Theory and Evidence. *The Journal of Finance*, 40: 777-790.
[14] Tversky, A., e Kahneman, D. (1981). The framing of decisions and the psychology of choice. *Science*, 211(4481), 453–458.
[15] Clarke, R. G., Krase, S. and Statman, M. (1994). Tracking Errors, Regret and Tactical Asset Allocation. *Journal of Portfolio Management*. Spring, 16-24.

valori fondamentali degli anni precedenti influenzano e guidano il loro processo decisionale.

1.2 La teoria dei mercati adattivi (AMH) come alternativa all'efficienza

Secondo la finanza comportamentale il mercato non è efficiente e che i partecipanti al mercato non sono razionali. Molte scoperte empiriche, come gli studi poco prima accennati, sono in contrasto con la teoria finanziaria standard dell'efficienza dei mercati. I critici della finanza comportamentale affermano che le queste teorie siano come una valigia di modelli *ad hoc* per spiegare singole anomalie senza mostrare una struttura generale (si veda, ad esempio, Fama, 1998). Un tentativo di sviluppare un tale quadro integrativo e quindi di dare una struttura a tutte queste evidenze, è stato proposto da Andrew W. Lo nel 2004 con l'ipotesi dei mercati adattivi (da *Adaptive Market Hypothesis*, abbreviato con AMH). L'AMH spiega l'avversione alla perdita, l'eccessiva reazione e altri pregiudizi comportamentali da parte delle reazioni degli investitori a un contesto di mercato in evoluzione. Con l'articolo "*The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective*" pubblicato nel 2004, Lo propone un nuovo quadro che riconcilia l'efficienza del mercato con le alternative comportamentali prima esposte, estendendo la nozione di "soddisfacimento" di Simon (1955) con le dinamiche evolutive. Come Lo spiega nel suo articolo, l'economista, psicologo e informatico Simon (1955) ha sostenuto che, poiché l'ottimizzazione è costosa e gli esseri umani sono naturalmente limitati nelle loro capacità computazionali, essi si concentrano sul "soddisfacimento", cioè un'alternativa all'ottimizzazione per cui gli individui compiono delle scelte semplicemente soddisfacenti, non necessariamente ottimali. In altre parole, gli individui sono limitati nel loro grado di razionalità, che è in netto contrasto con l'ortodossia per cui gli individui hanno razionalità illimitata.[16] Tuttavia, questa teoria è stata poco applicata nel mondo dell'economia e fu soggetta a diverse critiche, la più importante delle quali si chiedeva come è possibile determinare il

[16] Con questa idea, Simon ha ottenuto un premio Nobel nel 1978.

punto che separa il comportamento ottimale da quello soddisfacente. Una prospettiva evolutiva risponde e fornisce l'ingrediente mancante nel quadro di Simon: tali punti non sono determinati analiticamente, ma con un mix di selezione naturale e meccanismi di "prove ed errori".[17] Gli individui fanno scelte basate sull'esperienza passata e sulla loro migliore ipotesi su ciò che potrebbe essere ottimale, e imparano ricevendo un segnale positivo o negativo dai risultati. Se non ricevono tale segnale, non imparano. In questo modo gli individui si sviluppano l'arte della ricerca per risolvere le sfide economiche e, fintanto che tali sfide rimarranno stabili, quest'arte si adatterà per fornire soluzioni approssimativamente ottimali.

Sulla base di quanto appena descritto, Lo sostiene che gran parte dei controesempi alla razionalità economica (per esempio l'avversione alla perdita, l'eccessiva dipendenza, la reazione eccessiva, la contabilità mentale ecc.) sono coerenti con un modello evolutivo dove gli individui si adattano a un ambiente in evoluzione. In particolare, l'ipotesi dei mercati adattativi AMH può essere vista come una nuova versione di EMH, derivata da principi evolutivi.

Nonostante la natura piuttosto astratta e qualitativa dell'AMH, si possono derivare alcune implicazioni concrete. La prima implicazione è che nella misura in cui esiste una relazione tra rischio e rendimento, è improbabile che sia stabile nel tempo. Tale relazione è determinata dalle dimensioni e dalle preferenze relative di varie popolazioni nell'ecologia del mercato, nonché da aspetti istituzionali come l'ambiente regolamentare e le leggi fiscali. Poiché questi fattori si spostano nel tempo, è probabile che qualsiasi relazione rischio-rendimento ne sia influenzata. Un corollario di questa conseguenza è che il premio per il rischio azionario è variabile nel tempo. Questa non è un'idea così rivoluzionaria: anche nel contesto di un modello di equilibrio delle aspettative razionali, se le preferenze di rischio cambiano nel tempo, allora anche il premio per il rischio azionario deve variare. L'intuizione che l'AMH aggiunge è che le preferenze di rischio aggregate non sono costanti immutabili, ma sono modellate dalle forze della selezione

[17] Meccanismo *trial-and-error*, "prova e sbaglia", cioè tentare per valutare in seguito se il risultato ha prodotto l'effetto desiderato.

naturale.[18] Attraverso le forze della selezione naturale, la storia conta. Indipendentemente dal fatto che i prezzi riflettano pienamente tutte le informazioni disponibili, il particolare percorso che i prezzi di mercato hanno intrapreso negli ultimi anni influenza le attuali preferenze di rischio aggregato (Lo, 2004). Una seconda implicazione dell'AMH è che, diversamente dal classico EMH, di tanto in tanto esistono opportunità di arbitraggio che man mano che vengono sfruttati, scompaiono, ma nuove opportunità vengono continuamente create. Piuttosto che l'inesorabile tendenza verso una maggiore efficienza prevista dall'EMH, l'AMH implica dinamiche di mercato più complesse, con cicli e tendenze, bolle, crash. Una terza implicazione è che anche le strategie di investimento aumentano e diminuiscono, ottenendo buoni risultati in determinati ambienti e scarse prestazioni in altri ambienti. Contrariamente al classico EMH in cui le opportunità di arbitraggio sono messe in competizione eliminando la strategia progettata per sfruttarle, l'AMH implica che tali strategie potrebbero declinare per un certo periodo, per poi tornare utili quando le condizioni ambientali diventano più favorevoli a tali scambi. Una quarta implicazione è che l'innovazione è la chiave per sopravvivere. L'EMH classico suggerisce che determinati livelli di rendimenti attesi possono essere raggiunti semplicemente sopportando un grado sufficiente di rischio. L'AMH implica invece che la relazione rischio-rendimento varia nel tempo e che un modo migliore per raggiungere un livello coerente di rendimenti attesi è adattarsi alle condizioni mutevoli del mercato. Innovare significa sviluppare una molteplicità di capacità che si adattano a una varietà di condizioni ambientali, per cui i gestori degli investimenti hanno meno probabilità di estinguersi in seguito a rapidi cambiamenti delle condizioni. Infine, l'AMH ha una chiara implicazione per tutti i partecipanti ai mercati finanziari: la sopravvivenza è l'unico obiettivo che conta. Mentre la massimizzazione dei profitti, dell'utilità e il raggiungimento l'equilibrio generale sono certamente aspetti rilevanti dell'ecologia del mercato, il principio

[18] Ad esempio, i mercati statunitensi erano popolati da un significativo gruppo di investitori che non avevano mai sperimentato un vero mercato "orso". Lo scoppio della bolla tecnologica Dot.com del 2000 ha influenzato le preferenze di rischio della successiva popolazione di investitori. In questo contesto, la selezione naturale decide chi resta in gioco nel mercato: per esempio, gli investitori che hanno subito perdite sostanziali nella bolla tecnologica hanno maggiori probabilità di essere usciti dal mercato, lasciando una popolazione diversa di investitori (Lo, 2004).

organizzativo nel determinare l'evoluzione dei mercati e della tecnologia finanziaria è semplicemente la sopravvivenza (Lo, 2004).

Capitolo 2

Reinforcement Learning

2.1 Introduzione

L'intelligenza artificiale è lo studio di macchine dotate di un comportamento intelligente, comunemente inteso come capacità di imparare dall'esperienza, ispirandosi al comportamento e ai meccanismi umani dei cosiddetti esseri viventi superiori. Il *Machine Learning* è il campo che cerca di sviluppare agenti intelligenti tramite la tecnologia. Nell'era industriale ci si chiedeva "come posso progettare una macchina che lavora al posto mio?", mentre oggi, nell'era dell'informazione, la domanda diventa "come posso progettare una macchina che pensa al posto mio?".

Questo elaborato esamina alcune tecniche con le quali gli agenti possono imparare ad agire per risolvere i problemi di decisione sulla base della propria esperienza. Uno dei più significativi modi in cui imparano dipende dalla natura del feedback che ricevono. Per esempio, un'associazione tra situazioni, azioni e loro conseguenze potrebbe essere la seguente: "se il cielo è nuvoloso e non prendo l'ombrello allora è probabile che mi bagni". Semplice osservare che la conseguenza possibile di un'azione non permette di individuare quale sarebbe dovuta essere l'azione migliore. Senza associare una qualche forma di utilità al feedback non è possibile sapere quali cambiamenti potrebbero portare l'agente ad agire nel modo migliore. L'apprendimento senza questa utilità è chiamato apprendimento non supervisionato (in inglese *unsupervised learning*). Invece, un'indicazione nella forma "se il cielo è nuvoloso dovresti prendere l'ombrello" si chiama apprendimento supervisionato (in inglese *supevised learning*) e prevede un supervisore che è già a conoscenza di quale sia l'azione migliore in una certa situazione. Il supervisore può quindi dare suggerimenti per correggere le azioni dell'agente che le compie. Infine, se il feedback è dato come segnale positivo o

negativo, per esempio “Prima era nuvoloso. Non hai preso l’ombrello. Ora ti sei bagnato. Non va bene” allora l’agente sta imparando attraverso l’apprendimento per rinforzo (in inglese, *Reinforcement Learning,* abbreviato con RL). In questo caso l’apprendimento avviene con degli aggiustamenti sull’associazioone tra situazione e azione in modo tale da massimizzare i segnali positivi e minimizzare quelli negativi. A differenza delle altre due tecniche, nel Reinforcement Learning all’agente non viene mai detto quale azione è meglio prendere in una situazione, ma soltanto se e quanto questa azione sia utile. È compito dell’agente decidere alla fine quale azione scegliere sulla base delle informazioni che ha a disposizione. Questi tre tipi di apprendimento appena descritti sono i paradigmi del *Machine Learning* e questo elaborato si concentra solo sull’ultimo, il Reinforcement Learning.

2.2 Reinforcement Learning

Nel campo del *Machine Learning*, il Reinforcement Learning (d’ora in poi, anche abbreviato con RL) è una delle classi di algoritmi che si occupano di compiti decisionali sequenziali, ossia processi in cui un agente, per esempio un *software*, deve eseguire una sequenza di azioni per raggiungere alcuni obiettivi (Gao et al., 2000). Nel RL l’agente consiste in un programma intelligente che interagisce con l’ambiente in cui si trova, invece l’ambiente consiste in tutto quello che circonda l’agente e che non può controllare. Utilizzando il gioco degli scacchi come esempio, l’agente è il giocatore, mentre l’ambiente consiste nella scacchiera e il suo avversario. L’ambiente fornisce condizioni diverse a seconda della posizione delle pedine dei giocatori e influisce sulle mosse che l’agente compie. Queste condizioni vengono chiamate “stati”. Il ruolo dell’agente è quello di valutare lo stato in cui si trova e, sulla base di questo, scegliere un’azione da eseguire (nell’esempio muovere una pedina). Come conseguenza, l’azione modifica l’ambiente in una certa maniera e questa modifica viene comunicata all’agente attraverso un segnale numerico (chiamato *reward* o ricompensa) che misura con un numero la qualità dell’azione. In generale, l’obiettivo dell’agente è quello di selezionare le azioni per ottenere più ricompense positive possibili. All’agente

non viene detto che azioni intraprendere, ma deve scoprire da sé quali ottengono la massima ricompensa, attraverso l'esperienza. Questo meccanismo caratterizza il RL e si chiama *trial-and-error*, ossia "prova e sbaglia", ed è il meccanismo con cui l'agente esplora l'ambiente per ottenere più informazioni. Il RL è quindi un sistema dove l'agente apprende interagendo direttamente con l'ambiente, grazie ai segnali derivanti dalle sue azioni che gli permettono di capire quali azioni lo portano al risultato desiderato.

Per descrivere formalmente questo tipo di apprendimento solitamente si utilizzano dei processi matematici chiamati processi decisionali di Markov (abbreviati con l'acronimo MDP, dall'inglese *Markov decision process*), che forniscono un quadro matematico per modellare il processo decisionale. L'idea di base di questi processi è di descrivere il problema che l'agente si trova ad affrontare interagendo nel tempo con l'ambiente per raggiungere il suo obiettivo. Per farlo, il presupposto richiesto dai processi è che l'agente abbia un obiettivo già prestabilito, sia in grado di comprendere l'ambiente in cui si trova e possa compiere delle azioni che possono modificare lo stato dell'ambiente.

Una caratteristica del Reinforcement Learning, che non si trova nell'apprendimento supervisionato e non supervisionato, è il *trade-off* tra *exploration* ed *exploitation* ossia un bilanciamento tra esplorare nuove azioni e sfruttare quello che si è sperimentato. In generale, per ottenere una ricompensa positiva come conseguenza di un'azione, un agente di RL preferisce azioni che ha già provato nel passato e che ha scoperto essere proficue. Tuttavia, l'agente riceve solo il segnale numerico come conseguenza, non sa qual è l'azione migliore. Per scoprirlo deve provare altre azioni possibili che non ha mai compiuto in passato (*exploration*), così da capire quali restituiscono ricompense positive migliori (*exploitation*). In altre parole, l'agente deve sfruttare quello che già ha sperimentato per ottenere ricompense positive, ma deve anche tentare nuove azioni per fare scelte migliori nel futuro.

2.3 Elementi del Reinforcement Learning

Oltre all'agente e all'ambiente, ci sono quattro principali elementi in un sistema di RL: la *policy*, il segnale di *reward* o di ricompensa, la funzione valore e, in alcuni casi, un modello dell'ambiente (Barto e Sutton, 2018).

La descrizione del comportamento dell'agente si esprime attraverso la *policy*, la quale rappresenta una sorta di regola con cui l'agente associa un'azione a una data situazione, sulla base della sua percezione dello stato in cui si trova, cioè delle condizioni che caratterizzano l'ambiente. Viene quindi appresa dall'interazione che l'agente ha con l'ambiente. In particolare, la *policy* è il nucleo di un agente per l'apprendimento del rinforzo, nel senso che da sola è sufficiente a determinare il comportamento che può essere una funzione deterministica, il che significa che l'agente eseguirà sempre la stessa azione nello stesso stato oppure, come nell'elaborato, può essere una funzione stocastica, cioè una distribuzione di probabilità di tutte le azioni. L'obiettivo ultimo dell'agente è quello di determinare la cosiddetta "*policy* ottimale", ovvero trovare la *policy* che permetta all'agente di scegliere l'azione migliore ad ogni possibile stato dell'ambiente.[19]

In ogni periodo, come conseguenza dell'azione che l'agente sceglie di eseguire, l'ambiente invia all'agente un segnale numerico positivo o negativo chiamato *reward,* che rappresenta la qualità dell'azione. L'obiettivo dell'agente è quello di massimizzare la sommatoria dei *rewards* che riceve nel tempo, eventualmente scontati, in un'ottica di lungo periodo. In natura è possibile pensare al concetto di *reward* come delle esperienze di piacere e di dolore: per esempio, se un bambino mette il dito sul fuoco si brucia ma, grazie a questo segnale di dolore, in futuro sarà in grado di capire che non è un'azione profittevole. In più, i segnali di ricompensa sono anche gli strumenti utilizzati per modificare la *policy*: se infatti seguendo una *policy* si seleziona un'azione che non è profittevole (per esempio, perché restituisce un segnale negativo), allora verrà modificata la *policy*, così se in futuro l'agente si trovasse nella stessa situazione, selezionerebbe un'altra azione. In generale, i segnali di *reward* sono funzioni stocastiche dell'ambiente in cui l'agente si trova e dell'azione che compie.

[19] Questo punto viene descritto in modo approfondito successivamente con i processi di Markov.

Il meccanismo che raccoglie gli elementi appena descritti è rappresentato in Figura 2.1. Ad ogni periodo t $(t = 0,1,2,\dots)$, l'agente prende informazioni dell'ambiente, cioè dello stato in cui si trova, e sulla base di queste e della *policy* che lo caratterizza sceglie un'azione. L'esecuzione dell'azione fa sì che l'agente si trovi in un nuovo stato che, come conseguenza dell'azione, restituisce all'agente un segnale di *reward*.

Figura 2.1: Modello di interazione tra agente e ambiente.

La funzione valore è utile valutare la qualità di trovarsi in un certo stato. In altre parole, è una funzione che valuta a partire dallo stato in cui si trova l'agente, cosa si aspetta di ricevere nei periodi di tempo successivi. A differenza del segnale di *reward* che dà indicazione di cosa è profittevole nell'immediato, la funzione valore lo indica nel lungo periodo. Quindi, il valore di uno stato è la sommatoria dei *rewards* (eventualmente scontati) che un agente si aspetta di accumulare nel futuro, a partire da quello stato. In un certo senso i segnali di *reward* sono elementi primari mentre le funzioni valore secondari, poiché senza *rewards* non ci sono valori e l'obiettivo delle funzioni valore è di ottenere *rewards* sempre più positivi. Tuttavia, sono le funzioni valore che vengono utilizzate per valutare e prendere le decisioni. Infatti, le azioni sono scelte sulla base del giudizio dei valori: si compiono le azioni che ottengono il maggior valore non il maggiore *reward*, poiché ciò che conta è massimizzare questi ultimi in un'ottica di lungo periodo. Un punto critico dei modelli di RL sono le tecniche con cui le funzioni valore vengono stimate.

Infine, il modello per l'ambiente cerca di simulare il comportamento dell'ambiente in cui l'agente di trova. Per esempio, nel gioco degli scacchi l'ambiente è rappresentato dalla scacchiera e dall'avversario. Conoscere il

modello di questo ambiente significa conoscere le regole della scacchiera e il comportamento dell'avversario. In particolare, significa conoscere a priori come l'avversario risponderà ad ogni possibile mossa. Quindi, conoscere il modello significa sapere quali stati potrebbero seguire certe azioni e con che probabilità queste potrebbero avvenire. Dato uno stato e un'azione, il modello permette di prevedere quale stato seguirà e con quale segnale di *reward*. Tutte le informazioni relative ad un ambiente e disponibili all'agente vengono sintetizzate nel concetto di stato. Informalmente, lo stato dice all'agente come è l'ambiente in un determinato istante di tempo. Si noti che ai fini del RL non è necessario chiedersi come vengono costruiti gli stati, come arrivano all'agente o sono modificati, perché il RL è focalizzato solo sul problema decisionale. In altre parole, non riguarda come viene disegnato lo stato, ma quale azione viene intrapresa sulla base dello stato in cui ci si trova, qualsiasi esso sia.

2.4 Un esempio: Tic-Tac-Toe

Un esempio su come funziona il meccanismo di RL è descritto da Barto e Sutton (2018) attraverso il gioco del Tic-Tac-Toe, comunemente chiamato in italiano Tris. In questo gioco ci sono due giocatori e una griglia tre per tre. Uno alla volta, i giocatori riempiono uno spazio della griglia con il loro simbolo, generalmente un giocatore gioca X e l'altro gioca O. Uno dei due vince quando posiziona tre segni adiacenti (quindi un "tris") in una riga in qualsiasi orientamento, cioè verticale, orizzontale o diagonale, come rappresentato dal giocatore X nella Figura 2.2. Dato che in teoria un giocatore può giocare senza perdere mai, si assume ai fini dell'esempio che si stia giocando contro un giocatore "imperfetto", ossia che possa sbagliare mossa permettendo all'avversario di vincere. L'obiettivo è capire come costruire un agente in grado di identificare gli sbagli dell'avversario e massimizzare le possibilità di vincere.

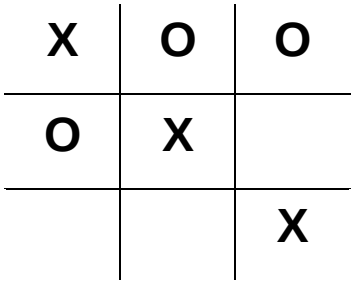

Figura 2.2: Esempio di Tic-Tac-Toe

Se si conoscessero tutte le probabilità con cui l'avversario compie una mossa per ogni stato del gioco, sarebbe possibile prevedere in partenza tutte le possibili mosse dell'avversario così da determinare una strategia che possa far vincere il giocatore. Tuttavia, si assume che questa informazione non sia disponibile come spesso avviene nella realtà. Allora, in alternativa, si può stimare questa informazione attraverso l'esperienza, giocando più partite contro l'avversario. Al meglio delle possibilità del giocatore, giocando più partite, si può capire qual è il comportamento che caratterizza l'avversario, cioè un modello dell'avversario, con il quale poi applicare la programmazione dinamica per ottenere una soluzione ottimale. Questo esempio è molto vicino al meccanismo di alcuni algoritmi presentati in questo elaborato.

Per introdurre le funzioni valore, si pensi ad una tabella di valori creata associando un valore ad ogni stato possibile della griglia del gioco, dove il valore è dato dalla probabilità di vincere contro l'avversario. Per un giocatore, uno stato A ha valore maggiore di uno stato B se il giocatore ha maggiore probabilità di vincere nello stato A piuttosto che nello stato B. Ad esempio, il giocatore che gioca X ha valore dello stato pari a 1 per tutti gli stati (le possibili configurazioni della griglia) dove ci sono tre X in una riga, poiché ha già vinto. Mentre, sempre dal punto di vista del giocatore X, per tutti gli stati in cui ci sono tre O in una riga la sua probabilità di vincere è 0 perché l'avversario ha già vinto.

Si giocano quindi diverse partite tra un giocatore e il suo avversario. Il giocatore per selezionare la mossa cerca di valutare tutti gli stati che potrebbero realizzarsi da ogni possibile mossa. La maggior parte del tempo, il giocatore compie la mossa secondo una logica *greedy*, cioè in alcuni casi seleziona la mossa

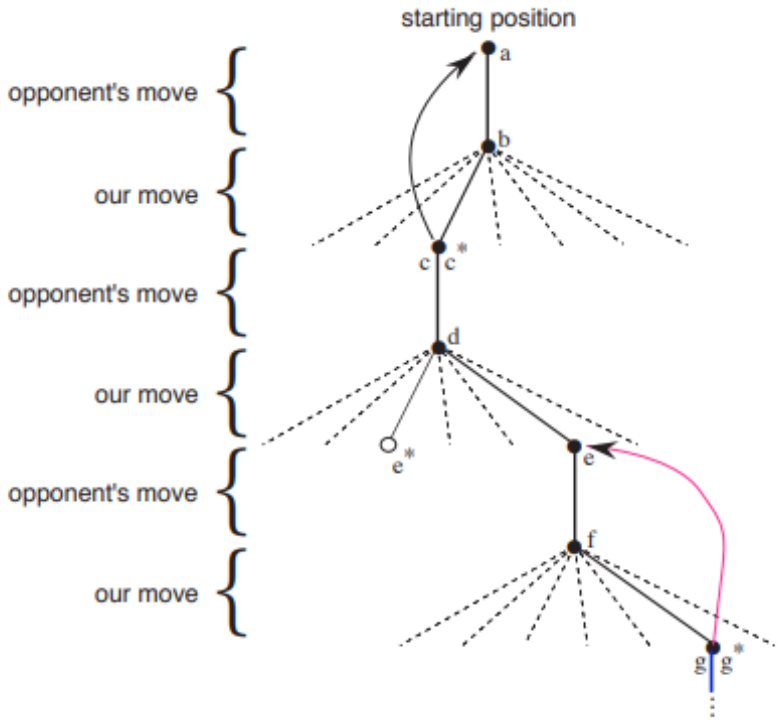

Figura 2.3: Sequenza di mosse del gioco Tic-Tac-Toe. La linea nera continua rappresenta le mosse che vengono intraprese; le linee tratteggiate sono le possibili azioni che il giocatore ha considerato ma non ha selezionato. La seconda mossa del giocatore, nella figura il secondo "our move", è una mossa esplorativa, nel senso che è stata presa anche se un'altra mossa di pari livello, quella che porta a e^*, ha maggior valore. Le mosse esplorative non comportano alcun apprendimento, ma ognuna delle nostre altre mosse lo fa, causando aggiornamenti come suggerito dalla freccia curva in cui i valori stimati vengono spostati sull'albero da nodi successivi ai precedenti. Fonte: Barto e Sutton, 2018.

che porta allo stato successivo con il maggior valore. In altri casi invece, giocherà delle mosse *random*, cioè delle mosse che non ha mai compiuto e che altrimenti non vedrebbe, al fine di esplorare e vedere a che risultati portano. Una sequenza di mosse compiute durante il gioco può essere rappresentata come in Figura 2.3. Mentre il giocatore sta giocando, aggiorna le probabilità di successo che associa allo stato quando ci si ritrova, cercando di stimare il più accuratamente possibile la probabilità di vincere da quello stato. Per fare questo, compie una sorta di aggiornamento del valore dello stato dopo ogni spostamento *greedy* nello stato prima dello spostamento, come suggerito dalle frecce della Figura 2.3. Si compie un backup, cioè se il passaggio da uno stato a quello successivo è rappresentato come nella figura dallo stato d allo stato f, il valore dello stato precedente d viene aggiornato sulla base del valore stimato dello stato successivo f. In formule, questo significa che se si indica con s il valore dello stato prima di aver compiuto

una mossa secondo la logica *greedy*, con s' il valore dello stato successivo, allora l'aggiornamento della stima del valore di s, indicata con $V(s)$, può essere formulata come

$$V(s) \leftarrow V(s) + \alpha\,[V(s') - V(s)]$$

dove α è un piccolo valore positivo chiamata *step-size parameter* che influenza di tasso di apprendimento. Questa è una regola esempio del metodo di apprendimento *Temporal-Difference* (spiegato nel capitolo 3), chiamato così poiché rappresenta una differenza, $[V(s') - V(s)]$, di stime della funzione valore in due tempi diversi. Se il parametro α viene ridotto nel modo corretto durante l'apprendimento, allora questo metodo converge alle vere probabilità di vincere di ogni stato. Inoltre, le mosse effettuate (eccetto quelle esplorative) sono in realtà le mosse ottimali (cioè le mosse che portano a più alta probabilità di vincere) contro l'avversario. In altre parole, il metodo converge ad una *policy* ottimale del gioco contro questo avversario.

Appare quindi evidente che un vantaggio della soluzione di apprendimento di rinforzo è che può ottenere una strategia ottimale (cioè che porta al miglior risultato) senza usare un modello dell'avversario e senza condurre una ricerca esplicita sulle possibili sequenze di stati e azioni futuri. Mentre questo esempio illustra alcune delle caratteristiche chiave del RL, è anche così semplice che potrebbe dare l'impressione che il RL sia più limitato di quanto non sia in realtà. Sebbene il Tic-Tac-Toe sia un gioco per due persone, il RL si applica anche nel caso in cui non vi siano avversari, vale a dire nel caso di un "gioco contro la natura". Per esempio, si può applicare al caso di un robot che cerca di uscire da una stanza. In più, il RL non si limita ai problemi in cui il comportamento si divide in episodi separati, cioè dei periodi come nell'esempio, dove la ricompensa si riceve solo alla fine, cioè alla fine di ogni partita. È altrettanto applicabile nei casi in cui il comportamento continua indefinitamente, per esempio quando si applica a sistemi di trading. Inoltre, è applicabile anche a problemi che non si dividono nemmeno in fasi temporali discrete, cioè $t = 0,1,2,\ldots$. Nell'esempio proposto, il gioco ha un set di stati finiti relativamente piccolo, mentre il RL può

essere usato quando il set di stati è molto grande o addirittura infinito. Ad esempio, Gerry Tesauro (1992, 1995) ha combinato l'algoritmo sopra descritto con una rete neurale artificiale, un modello matematico ispirato al meccanismo dei neuroni, per imparare a giocare a Backgammon, un gioco complesso che ha circa 10^{20} stati. Con così tanti stati è possibile sperimentare meno di una piccola parte di essi. Tuttavia, la rete neurale fornisce al programma la capacità di generalizzare in base alla sua esperienza basata su un tipo di apprendimento supervisionato, cioè la capacità di selezionare in nuovi stati le mosse in base alle informazioni salvate da stati simili affrontati in passato. Il modo in cui un sistema di RL può funzionare in problemi con insiemi di stati così vasti è legato a quanto adeguatamente può generalizzare dall'esperienza passata. Questo metodo che utilizza reti neurali in letteratura prende il nome di apprendimento profondo (dall'inglese *Deep Learning*).

Infine, è utile sottolineare che il giocatore nel Tic-Tac-Toe è in grado di "guardare avanti" e conoscere alcuni degli stati che derivano da ciascuna delle sue possibili mosse. Per fare questo, deve avere un modello del gioco che permette di pensare a come il suo ambiente sarebbe cambiato in risposta alle mosse possibili. Molti problemi di apprendimento sono così, ma in altri manca anche un modello a breve termine degli effetti delle azioni. Il RL può essere applicato in entrambi i casi. Non è richiesto alcun modello, poiché può essere appreso, come illustrato dall'esempio. Infatti, ci sono metodi di apprendimento di rinforzo che non necessitano di alcun tipo di modello ambientale. I sistemi *model-free* non possono nemmeno pensare a come i loro ambienti cambieranno in risposta a una singola azione, come per il giocatore nel Tic-Tac-Toe rispetto al suo avversario. Poiché i modelli devono essere ragionevolmente precisi per essere utili, i metodi *model-free* possono avere vantaggi rispetto a metodi più complessi quando il nodo nella risoluzione di un problema è la difficoltà di costruire un modello ambientale sufficientemente accurato.

2.5 Processi decisionali finiti *a là* Markov

Come già anticipato, i problemi di RL vengono formalizzati con i processi decisionali alla Markov, processi utilizzati più in generale per la formalizzazione dei problemi decisionali sequenziali dove si esegue una sequenza di azioni per raggiungere alcuni obiettivi e dove le azioni influenzano non solo le ricompense immediate, ma anche gli stati futuri e, tramite questi, le ricompense future.

2.5.1 Formalizzazione dei processi

Un processo decisionale alla Markov (d'ora in poi indicato anche con MDP, da *Markov Decision Process*) è un modello di un agente che interagisce in modo sincrono con il suo ambiente. L'agente prende come input lo stato dell'ambiente e genera come output delle azioni, che a loro volta influenzano l'ambiente (Figura 2.4). Nel quadro del MDP si presume che, sebbene possa esserci una grande incertezza sugli effetti delle azioni, non vi è mai alcuna incertezza sulla conoscenza dello stato attuale da parte dell'agente: esso ha capacità percettive complete e perfette. L'assunzione di "sensori" perfetti è fondamentale: dato che l'agente è in grado di percepire direttamente tutti gli aspetti dello stato corrente, che potrebbero essere necessari per stimare la probabilità dello stato successivo data la sua azione, non è necessario che l'agente conservi alcuna memoria delle

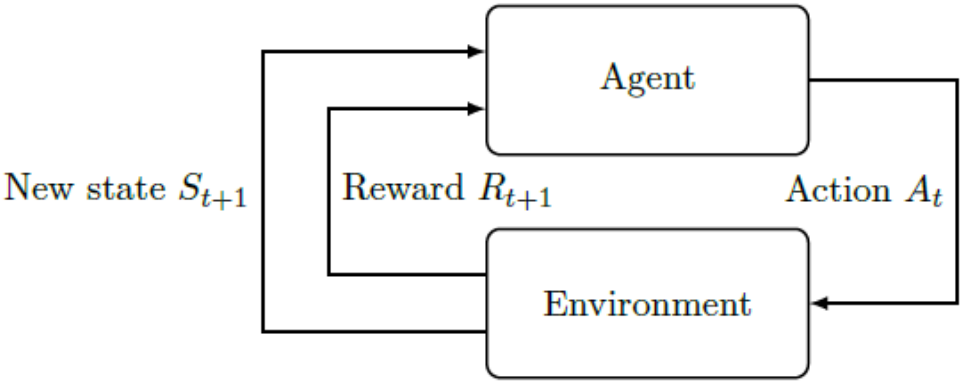

Figura 2.4: Meccanismo di un MDP. Fonte: Corazza et al., 2019.

sue azioni o stati passati per prendere decisioni ottimali. Questi processi si articolano in stati, azioni, transizioni stato-azione e una funzione di *reward*.

Sebbene in generale si possano avere infiniti stati e azioni, la discussione per tutto l'elaborato si limita a problemi dove stati e azioni sono finiti.

Formalmente, un set di azioni è definito come un set finito $\{A_1, \dots, A_K\}$ di dimensione K, cioè dal tempo $t = 1, \dots, K$. Il set di azioni che può essere applicato in un particolare stato $S_t \epsilon \mathcal{S}$ si indica $A(S_t)$, dove $A(S_t) \subseteq \mathcal{A}$. L'agente è il sistema responsabile di interagire con l'ambiente per prendere le decisioni, per scegliere quale azione compiere ad un determinato stato. Egli dovrebbe agire in modo tale da massimizzare la ricompensa nel lungo termine.

Gli agenti operano in un ambiente, il cui stato contiene le informazioni rilevanti che l'agente può utilizzare per compiere le sue scelte. Un set di stati è definito come un set finito $\{S_1, \dots, S_N\}$ di dimensione N, con $t = 1, \dots, N$.

Quando l'agente compie un'azione influenza l'ambiente, cioè il sistema compie una transizione dallo stato S_t ad un nuovo stato S_{t+1} sulla base di una distribuzione di probabilità su tutte le transizioni possibili. La transizione da uno stato all'altro è rappresentata dalla funzione di transizione T. Formalmente è definita come $T: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$, ovvero $T(S_t, A_t, S_{t+1})$, ed è la probabilità di ottenere lo stato S_{t+1} dopo aver compiuto l'azione A_t essendo nello stato S_t. Dunque, è necessario per ogni azione A_t e per ogni stato S_t e S_{t+1} che $T(S_t, A_t, S_{t+1}) \geq 0$ e $T(S_t, A_t, S_{t+1}) \leq 1$. Inoltre, $\sum_{s' \epsilon S} T(S_t, A_t, S_{t+1}) = 1$ per ogni azione A_t e per ogni stato S_t e S_{t+1}. Per potersi riferire all'ordine con il quale le azioni vengono compiute o lo stato in cui ci si trova, è utile utilizzare anche la notazione con riferimento temporale, cioè S_t e S_{t+1} e A_t dove $t = 1, 2, \dots$.

La funzione di *reward* definisce le ricompense che si ottengono per aver raggiunto uno stato o per aver compiuto una certa azione essendo in un dato stato. Si noti che con il termine ricompensa si fa riferimento ad un mero feedback scalare, dunque ha sia una connotazione positiva che negativa (premio e penalità). La funzione di *reward*, indicata con R, si definisce come $R: \mathcal{S} \rightarrow \mathbb{R}$ e indica il numero reale che corrisponde al segnale di feedback conseguente alla transizione tra stati. Vi sono altre due definizioni della funzione di *reward*, in base a come viene valutata: la prima si indica $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ e sottolinea il punto di vista della ricompensa per aver compiuto un'azione, la seconda definisce $R: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ come ricompensa di una particolare transizione tra stati.

La descrizione del comportamento dell'agente si esprime attraverso la *policy*. Formalmente una *policy*, indicata con π, è la funzione $\pi: \mathcal{S} \to \mathcal{A}$ che attribuisce per ogni stato $S_t \, \epsilon \, \mathcal{S}$ un'azione $A_t \, \epsilon \, A(S_t)$. Partendo da uno stato iniziale S_0, la *policy* π suggerisce l'azione $A_0 = \pi(S_0)$. Sulla base della funzione di transizione T e della funzione di *reward* R, avviene una transizione dallo stato S_0 al nuovo stato S_1 con probabilità $T(S_0, A_0, S_1)$ e una ricompensa $R_0 = R(S_0, A_0, S_1)$. Un concetto importante e che si trova spesso lungo l'elaborato è la *greedy-policy*, letteralmente una *policy* "avida". Una *greedy-policy* significa che l'agente esegue la *policy* che porta all'azione che si ritiene produca il premio più alto previsto. Ovviamente, una tale *policy* non consente all'agente di esplorare, secondo la logica del *trade-off* di esplorazione e sfruttamento alla base del RL, come precedentemente introdotto. Per consentire qualche esplorazione, viene spesso utilizzata una *policy* ε-*greedy*: viene selezionato un numero (chiamato ε) nell'intervallo di $[0,1]$ e, prima di selezionare un'azione, un numero casuale sempre nell'intervallo di $[0,1]$. Se quel numero è maggiore di ε, viene selezionata l'azione *greedy*, ma se è inferiore, viene selezionata un'azione casuale. Si noti che se $\varepsilon = 0$, la *policy* diventa la *greedy* e se $\varepsilon = 1$, esplora sempre.

Mettendo insieme tutti questi elementi, è possibile definire un processo decisionale alla Markov come una tetrupla $\langle S, A, T, R \rangle$ nella quale S è un set finito di stati, A è un set finito di azioni, T è una funzione di transizione definita come $T: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ e R è una funzione di rinforzo definita come $R: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. Un sistema si dice markoviano se il risultato di un'azione non dipende dagli stati e dalle azioni precedenti, ma dipende solamente dallo stato attuale, cioè:

$$P(S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, \dots) = P(S_{t+1}|S_t, A_t) = T(S_t, A_t, S_{t+1}). \qquad (2.1)$$

L'idea alla base dei processi di Markov è che lo stato attuale S_t dia abbastanza informazioni per prendere una scelta ottimale, indipendentemente dagli stati e dalle azioni che hanno preceduto S_t.

Riepilogando, l'agente e l'ambiente interagiscono a tempi discreti, l'agente selezionando le azioni mentre l'ambiente rispondendo a queste azioni con le ricompense e presentando nuove situazioni. L'agente ha l'obiettivo di

massimizzare la funzione valore attraverso la scelta di azioni nel tempo. Più specificamente, l'agente e l'ambiente interagiscono ad ogni sequenza finita di periodi temporali discreti, $t = 0,1,2, \ldots$. Ad ogni tempo t, l'agente riceve una rappresentazione dell'ambiente con lo stato $S_t \in \mathcal{S}$, sulla base di questo sceglie di compiere un'azione $A_t \in A(S_t)$. Al periodo successivo, come conseguenza della scelta dell'azione, l'agente riceve un segnale di *reward* numerico $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$ e si ritrova in un nuovo stato S_{t+1}. Quindi si genera una sequenza come la seguente:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots .$$

Si noti che la ricompensa che si ottiene come conseguenza dell'azione allo stato iniziale S_0, viene ricevuta dall'agente nel periodo successivo e, infatti, è indicata con R_1. Con questa notazione si vuole sottolineare che la ricompensa viene generata come conseguenza della transizione dallo stato S_t allo stato S_{t+1}.

2.5.2 Criteri e ottimalità

L'obiettivo dell'agente è massimizzare l'ammontare totale di *rewards* che riceve, ossia i *rewards* cumulati nel lungo periodo. In generale si vuole massimizzare il rendimento atteso G_t, inteso come una qualche funzione dei *rewards* che si ricevono dopo il periodo t. Nel caso più semplice si definisce come somma dei *rewards* futuri, cioè:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T \, . \tag{2.2}$$

La maggior parte degli approcci in letteratura si basa su tre modelli di ottimalità a lungo termine: l'orizzonte finito, l'orizzonte infinito e la ricompensa media[20] (Wiering e Van Otterlo, 2012).

[20] Questa classificazione non intende esaurire la gamma di possibili modelli, ma fornisce un vocabolario per esprimere funzioni e strutture presenti nei capitoli successivi.

Nel modello ad orizzonte finito, un agente considera un periodo finito T e ottimizza le sue ricompense attese lungo questo arco di tempo, come descritto sopra, cioè:

$$\mathbb{E}[G_t] = \left[\sum_{h=t+1}^{T} R_h\right]. \tag{2.3}$$

L'agente al primo passo sceglie l'azione che ottimizza i prossimi T passi, poi al secondo ottimizza i $T-1$ passi successivi, e così via. Questo metodo è utile in situazioni che sono naturalmente a tempo finito, ovvero quando le interazioni ambiente-agente sono per natura divise in periodi, chiamati *episodi*, come nel gioco degli scacchi o del tris dove le interazioni giocatore-ambiente si ripetono in periodi.

Però, nella maggior parte dei casi le interazioni non sono divise in episodi in modo naturale, ma sono continue e senza limite. In questo caso, si avrebbe una sorta di formulazione come quella sopra, con la differenza che $T \to \infty$ e, di conseguenza, il valore atteso finale potrebbe a sua volta essere infinito. Per ovviare a questi problemi si utilizza la nozione di modello ad orizzonte scontato e infinito, nel quale le ricompense future vengono scontate sulla base di quanto distanti nel tempo si ricevono:

$$\mathbb{E}[G_t] = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \left[\sum_{h=t+1}^{\infty} \gamma^t R_h\right] \tag{2.4}$$

dove γ è il fattore di attualizzazione e $0 \leq \gamma < 1$. Il fattore di attualizzazione può essere interpretato in modi diversi, per esempio come tasso di interesse o come probabilità di vita nei periodi futuri. Per $\gamma < 1$, le ricompense che si ricevono più tardi nel tempo sono scontate più di quelle vicine, così il valore atteso può convergere ad un numero finito, nonostante l'orizzonte temporale sia infinito. Per esempio, se si immagina di ricevere ad ogni periodo un *reward* costante pari a +1, allora la somma convergerà ad un numero finito e pari a

$$G_t = \sum_{t}^{\infty} \gamma^t = \frac{1}{1-\gamma}. \tag{2.5}$$

Per $\gamma = 0$, l'agente è "miope": l'oggetto dell'agente in questo caso è l'apprendimento attraverso la sola scelta di A_t, ossia massimizza solo quanto riceve in $t + 1$. Quando invece il fattore γ si avvicina ad 1, l'agente dà maggiore considerazione ai *rewards* futuri. Per i *rewards* vale la seguente relazione:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \\ &= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \cdots) \\ &= R_{t+1} + \gamma G_{t+1}. \end{aligned} \tag{2.6}$$

Il terzo modello di ottimalità, la ricompensa media, massimizza le ricompense future medie:

$$G_t = \lim_{h\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{h=t+1}^{T} R_h\right]. \tag{2.7}$$

Con quest'ultimo modello di ottimalità, per periodi lunghi o infiniti non è possibile distinguere tra due *policies*, dove una ottiene più ricompense all'inizio e l'altra che le ottiene in periodi più distanti, rappresentando uno svantaggio nell'uso di questo modello.

Questi modelli possono essere considerati come la funzione che si desidera massimizzare durante l'apprendimento. Tuttavia, il criterio di ottimalità è un concetto più ampio che non si limita alla funzione che si vuole ottimizzare. L'ottimalità riguarda anche l'efficienza con cui gli algoritmi operano, ossia una valutazione del modo più o meno efficace con cui cercano di ottenere soluzioni ottimali. Questo concetto più generale può riassumersi brevemente in tre aspetti.

Il primo riguarda la capacità dell'agente di raggiungere una strategia ottimale, cioè la strategia migliore fra tutte le possibili strategie. In altre parole, ci si chiede se ci sia un modo per assicurarsi che il processo di apprendimento converga ad una *policy* ottimale. Per alcuni modelli esistono delle proprietà che permettono di dimostrarlo, per altri invece non è possibile.

Il secondo concetto riguarda la velocità di convergenza alla soluzione: quanti calcoli sono necessari per raggiungere l'ottimo? Nella pratica, per esempio, un elicotterista non può permettersi gli stessi errori nell'apprendimento di un sistema di trading o di un robot di piccole dimensioni.

Infine, l'ultimo aspetto riguarda quante ricompense non ottenute si realizzano rispetto alla soluzione ottimale.

2.5.3 *Policies* e funzioni valore

Gli algoritmi di RL prevedono di stimare le funzioni valore degli stati, le quali indicano il livello di utilità dell'agente di trovarsi in un certo stato (ovvero quanto valore abbia compiere una certa azione essendo in un certo stato). La nozione di valore si intende come valore atteso della somma del *reward* attuale e dei *rewards* futuri. Come già anticipato, le funzioni valore dipendono dalla *policy*, poiché gli stati in cui l'agente si troverà, quindi il valore di questi stati, dipende dal comportamento che l'agente mette in atto.

Formalmente una *policy* è una mappatura dallo stato alle probabilità che venga selezionata ciascuna azione. Per una politica π al tempo t, $\pi(a|s)$ è la probabilità che si scelga $A_t = a$ essendo lo stato $S_t = s$.[21] I metodi di RL specificano come la *policy* dell'agente cambia in funzione della sua esperienza. La funzione valore di uno stato s con *policy* π, indicata $v_\pi(s)$, è il valore atteso delle ricompense partendo dallo stato s e applicando la *policy* π. Per i MDP e utilizzando il modello ad orizzonte infinito dei *rewards*, $v(s)$ si esprime formalmente come[22]:

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s\right], \tag{2.8}$$

[21] La nomenclatura spesso viene semplificata con le lettere minuscole. In particolare si precisa che $S_t = s$, $A_t = a$, $S_{t+1} = s'$, $A_{t+1} = a'$, ecc.
[22] Con la notazione $\mathbb{E}_\pi$, si precisa che il valore atteso è calcolato applicando la policy π.

per ogni $s \epsilon S$. Analogamente, si definisce la funzione valore stato-azione, $q_\pi(s,a)$, come il valore atteso delle ricompense partendo dallo stato s, compiuta l'azione a e seguendo la *policy* π per tutto il processo:

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a\right]. \quad (2.9)$$

La differenza tra le due funzioni è che nel primo caso, $v(s)$, si valuta la qualità di trovarsi in un certo stato, mentre nella seconda, $q_\pi(s,a)$, si valuta dal punto di vista della combinazione della coppia stato-azione, quindi il valore di trovarsi nello stato s poiché si è compiuta l'azione a.

Una proprietà fondamentale delle funzioni valore è che soddisfano delle particolari relazioni ricorsive. È possibile provare che, per ogni *policy* π e per ogni stato s, la seguente relazione resta valida durante le transizioni tra gli stati:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s] \quad \text{da (2.6)} \\ &= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)\left[r + \gamma \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s']\right] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma v_\pi(s')\right] \quad (2.10) \end{aligned}$$

dove a' e s' sono l'azione e lo stato successivi a a e s rispettivamente, mentre r è il *reward* che deriva dalla scelta a presa allo stato s. L'espressione finale è una sommatoria di funzioni di tre variabili, a, s' ed r, che stabilisce che il valore dello stato di partenza è uguale al valore scontato dei possibili stati futuri, più il prossimo *reward* atteso. Per ogni tripletta si considerano le corrispondenti probabilità[23], con cui si pesano le quantità nelle parentesi quadre. Infine, i valori risultanti si sommano per ottenere il valore atteso. In altre parole, l'equazione indica che il valore atteso di uno stato si definisce in termini di valori futuri attesi, pesati per le rispettive probabilità e un fattore di sconto. Questa equazione prende

[23] Si noti che le espressioni $\pi(a|s)$ e $p(s', r|s, a)$ rappresentano le probabilità della funzione di transizione T.

il nome di equazione di Bellman (1957) ed esprime la relazione tra il valore di uno stato e il valore dei suoi possibili stati successivi. In particolare, indica che il valore dello stato iniziale deve essere uguale al valore (scontato) dello stato successivo atteso, più il premio atteso lungo il percorso. È possibile dimostrare che per MDP finiti esiste un'unica soluzione finita. Si noti che diverse *policy* π possono avere la stessa funzione valore ma, data una *policy* π, $v_\pi(s)$ è unica.

L'obiettivo di ogni MDP è quello di trovare la *policy* ottimale, cioè la *policy* che permette di ricevere più compensi. Dunque, significa massimizzare la funzione valore $v_\pi(s)$ descritta dall'Equazione 2.1. Formalmente, una *policy* ottimale π_* è tale che $v_{\pi_*}(s) \geq v_\pi(s)$ per ogni $s \, \epsilon \, S$. In termini di equazione di Bellman, la soluzione ottima (semplificata con $v_* = v_{\pi_*}$) può essere riscritta come segue:

$$v_*(s) = \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} | S_t = s] \qquad \text{Da (2.10)}$$

$$= \max_a \mathbb{E}[R_{t+1} + \gamma v^*(S_{t+1}) | S_t = s]$$

$$= \max_a \sum_{s',r} p(s', r|s, a) \, [r + \gamma v_*(s')]. \qquad (2.11)$$

Questa espressione è chiamata equazione di ottimalità di Bellman ed indica che il valore ad uno stato con una *policy* ottimale deve essere uguale al valore atteso delle ricompense ottenute con la migliore azione possibile in quello stato. Per esprimere l'azione ottimale, data la funzione valore v_*, si può utilizzare la seguente forma:

$$\pi_*(s) = \arg \max_{a \epsilon A} \sum_a \pi(a, s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_{\pi_*}(s') \right]. \qquad (2.12)$$

Questa *policy* viene chiamata *greedy-policy*, anche indicata con $\pi_{greedy}(V)$, poiché seleziona l'azione che porta allo stato con il maggior valore. Analogamente, l'equazione di ottimalità di Bellman per $q_*(s, a)$ è

$$q_*(s, a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q^*(S_{t+1}, a') | S_t = s, A_t = a \right] \qquad (2.13)$$

$$= \sum_{s',r} p(s', r|s, a) \left[r + \gamma \max_{a'} q_*(s', a') \right].$$

Per ogni coppia stato-azione (s, a), questa relazione esprime il valore atteso per aver scelto l'azione a allo stato s e seguendo una *policy* ottimale. Come per l'Equazione 2.12, la selezione dell'azione ottimale può essere espressa in termini di funzione valore stato-azione:

$$\pi_*(s) = \arg \max_{a \epsilon A} q_*(s, a), \tag{2.12}$$

vale a dire, l'azione migliore è l'azione che porta al maggiore valore della funzione valore conseguibile in base ai possibili stati successivi che risultano eseguendo tale azione essendo nello stato s. Infine, è possibile esprimere la relazione tra $v_*(s)$ e $q_*(s, a)$ nel seguente modo

$$q_*(s, a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \tag{2.13}$$

che esprime la funzione valore stato-azione ottimale q_* in termini di funzione valore v_*.

Risolvere esplicitamente l'equazione di ottimalità di Bellman fornisce una strada per trovare una *policy* ottimale e, di conseguenza, per risolvere il problema di RL. Tuttavia, questa soluzione si basa su almeno tre ipotesi che raramente sono vere nella realtà:

- le dinamiche dell'ambiente sono note e accurate, ossia la distribuzione di probabilità delle funzioni T e R sono entrambe note;
- si dispone di risorse computazionali sufficienti per completare il calcolo della soluzione;
- vale la proprietà di Markov, per la quale il risultato di un'azione non dipende dagli stati e dalle azioni precedenti, ma dipende solamente dallo stato attuale.

Generalmente, nella realtà vengono violate varie combinazioni di questi presupposti. Ad esempio, sebbene la prima e la terza ipotesi non presentino problemi per il gioco del Backgammon, la seconda costituisce un importante

limite. Poiché il gioco ha circa 10^{20} stati, i computer impiegherebbero troppo tempo per risolvere l'equazione di Bellman che valuterebbe tutte le combinazioni stato-azione. Invece, il RL può ovviare a questi ostacoli cercando delle soluzioni che approssimano le funzioni valore.

Capitolo 3

Algoritmi fondamentali di Reinforcement Learning

Il problema del processo decisionale sequenziale è ampiamente studiato nell'intelligenza artificiale ed in letteratura si trovano diversi modelli che hanno l'obiettivo di risolverlo, tra i quali il Reinforcement Learning introdotto nel capitolo precedente. Ora che sono stati definiti gli strumenti con i quali si rappresenta il RL, cioè i processi di Markov, si analizzano alcune delle tecniche di soluzione di questi processi, vale a dire degli algoritmi di diverse famiglie (programmazione dinamica, metodi Monte Carlo e, in particolare, il RL) che consentono di risolvere un processo di Markov, quindi di raggiungere una *policy* ottimale.

La distinzione principale a cui fanno riferimento gli algoritmi che affrontano i problemi di RL è quella tra metodi *model-based* e *model-free*. I metodi *model-based* assumono che un modello per l'ambiente sia noto a priori, cioè si conosca la distribuzione di probabilità delle funzioni di transizione. Appartiene a questa categoria la prima classe di metodi presentati, chiamata programmazione dinamica (dall'inglese *Dynamic Programming*) appunto caratterizzata dal fatto che un modello è disponibile e viene utilizzato nei processi. Al contrario, la classe dei metodi *model-free* è caratterizzata dall'assunzione per cui non è dato un modello del MDP già noto a priori e, anche qualora lo fosse, non è necessario per raggiungere la soluzione del processo decisionale. Al suo posto questi algoritmi si affidano all'interazione che l'agente ha con l'ambiente che produce un campione di *rewards* e di transizioni tra stati, con cui stimare le funzioni valore. Quindi, sono algoritmi che apprendono dall'esperienza diretta dell'agente. Di questa classe vengono presentati quelli basati sui metodi Monte

Carlo e i metodi di apprendimento per differenze temporali (dall'inglese *Temporal-Difference Learning*).

Gli algoritmi di apprendimento per differenze temporali combinano concetti caratteristici della programmazione dinamica e dei metodi Monte Carlo, per cui il capitolo è strutturato in modo da definire gli elementi di programmazione dinamica e Monte Carlo per poi presentare l'apprendimento di per differenze temporali. Nel prossimo paragrafo, si presenta il principio di *Generalized Policy Iteration* (GPI) alla base di tutti i processi.

3.1 Il principio di Generalized Policy Iteration (GPI)

Esiste un principio comune a tutte le classi di algoritmi che verranno presentate che prende il nome di *Generalized Policy Iteration* (GPI). Questo principio si articola in due fasi, ossia la valutazione della *policy* e il suo miglioramento[24], le quali si alternano nei processi decisionali interagendo tra loro per ottenere la *policy* ottimale e la relativa funzione valore. La valutazione della *policy* permette di stimare il valore della funzione valore v_π associata a una *policy* π fissata. L'obiettivo di questa valutazione è ottenere informazioni sulla *policy* da utilizzare come input in una fase successiva. In questa seconda fase, invece, l'obiettivo è di ottenere un miglioramento della *policy*, cioè la determinazione di nuove azioni da eseguire in particolari stati che performano meglio delle azioni che la *policy* attuale propone (Wiering, 2005). Il meccanismo è rappresentato nella Figura 3.2. Il modo in cui si applicano le fasi e il grado di separazione tra le due caratterizzano gli algoritmi o la classe a cui appartengono. Per esempio, nella programmazione dinamica descritta nel prossimo paragrafo, si presenta un algoritmo nel quale le due fasi sono nettamente separate ed uno in cui sono in parte sovrapposte. Se sia la valutazione che il miglioramento della *policy* si stabilizzano, cioè non generano più cambiamenti, allora la funzione valore e la *policy* hanno raggiunto il loro valore ottimale, cioè il migliore tra le possibili alternative. La funzione valore viene determinata sulla base della *policy* corrente e la *policy* si determina quando è *greedy* rispetto alla funzione valore corrente, cioè

[24] Dall'inglese, rispettivamente, *policy evaluation* e *policy improvement.*

esegue l'azione che si ritiene produca la ricompensa prevista più alta. Pertanto, entrambi i processi si stabilizzano solo quando è stata trovata una *greedy-policy* rispetto alla propria funzione di valutazione. Questo significa che vale l'equazione di ottimalità di Bellman (2.11), quindi sia la *policy* che la funzione valore sono ottimali.

Si potrebbe anche pensare all'interazione tra le due fasi in termini di due obiettivi, come rappresentati dalle due linee non ortogonali nello spazio bidimensionale nal diagramma a destra nella Figura 3.2. Sebbene sia più complicata di così, il diagramma suggerisce una semplificazione di cosa avviene nella pratica. Il principio di GPI, alternando valutazione e miglioramento della *policy*, guida la funzione valore o la *policy* verso una delle linee che rappresentano, rispettivamente, la soluzione ottima della *policy evaluation* e della *policy improvement;* cioè nel diagramma $V = V^{\pi}$ è l'obiettivo della valutazione della policy e $\pi = greedy(V)$ è l'obiettivo della fase di miglioramento. Muoversi direttamente verso un obiettivo provoca un allontanamento dall'altro. Per esempio, la fase di miglioramento della *policy* parte dalla funzione valore v_{π} e permette di ottenere una nuova *policy* (π') migliore rispetto a prima, che però non è più rappresentata correttamente con $v(\pi)$ e quindi si è allontanata dall'obiettivo della fase di valutazione. Tuttavia, il processo nell'insieme si avvicina all'obiettivo generale di ottimalità. Le frecce in questo diagramma corrispondono alla dinamica delle due fasi che guidano il sistema fino al raggiungimento completo di uno dei due obiettivi. Si usa quindi il termine *Generalized Policy Iteration* (GPI) per fare

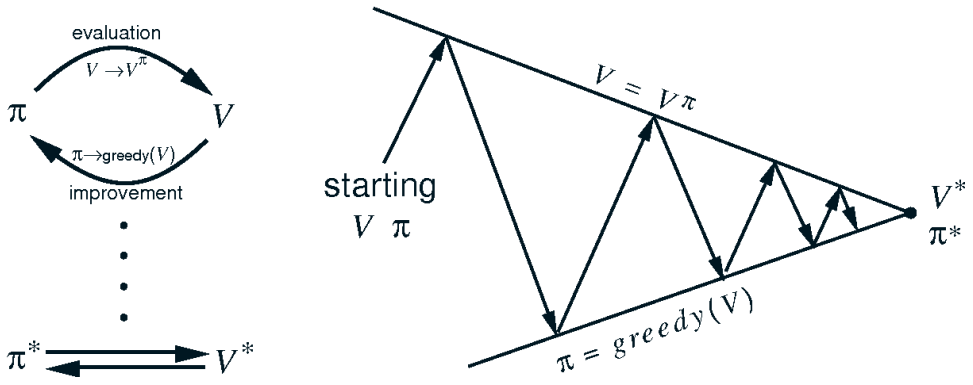

Figura 3.2: A sinistra, la fase di valutazione della policy (evaluation) stima v_{π}, cioè le prestazioni della policy. La fase di miglioramento della policy (improvement) migliora la policy π sulla base delle stime di v_{π}. A destra, la convergenza di π e V verso i loro valori ottimali. Fonte: Wiering, 2005.

riferimento al concetto generale di far interagire il processo di valutazione e di miglioramento della *policy*, indipendentemente dalla separazione e da altri dettagli delle due fasi.

3.2 Programmazione dinamica

La programmazione dinamica è una classe di algoritmi *model-based*, cioè caratterizzati da un modello per l'ambiente noto. Nella pratica è un'assunzione difficile da garantire, di conseguenza il suo uso è limitato. Tuttavia, gli algoritmi di programmazione dinamica restano importanti come supporto per presentare i come si comportano gli algoritmi in mancanza del modello. Questi ultimi infatti, sono tentativi di ottenere lo stesso risultato della programmazione dinamica senza assumere un modello.

Gli algoritmi di programmazione dinamica che vengono presentati, come già anticipato, si differenziano per il grado di separazione delle fasi del GPI: si presentano il *Policy Iteration* (Howard, 1960) e il *Value Iteration* (Bellman, 1957), dove il primo presenta le due fasi chiaramente separate, mentre nel secondo sono leggermente integrate tra loro.

Si assuma per l'esposizione che l'ambiente sia un processo MDP finito, ossia si assuma che i set di stati, azioni e *rewards* siano finiti e che le loro dinamiche siano descritte da un set di probabilità $p(s', r|s, a)$ note, per ogni $s, s' \epsilon S, a \epsilon A, r \epsilon R$.

3.2.1 Prima fase: la valutazione della policy

L'obiettivo della valutazione della *policy*, anche chiamato problema di previsione, consiste nel determinare una funzione valore v_π per una *policy* π fissata. Dal capitolo 2 si ricorda la definizione della funzione valore per ogni stato $s \epsilon S$,

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] \qquad \text{da (2.6)}$$

$$= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s]$$
$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \left[r + \gamma E_\pi[G_{t+1}|S_{t+1} = s']\right]$$
$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right] \quad (3.1)$$

dove $\pi(a|s)$ è la probabilità di compiere l'azione a trovandosi il sistema allo stato s con *policy* π. Inoltre, la funzione valore v_π esiste ed è unica finché $\gamma < 1$. Se la dinamica dell'ambiente è perfettamente nota, l'Equazione 3.1 è un sistema di $|S|$ equazioni con $|S|$ incognite, ossia $v_\pi(s)$ per ogni $s \epsilon S$, quindi è un sistema risolvibile direttamente seppur tedioso a livello di calcolo. In alternativa, a questo problema si può ovviare con una procedura iterativa utilizzando minori risorse di calcolo. Si considera una sequenza di funzioni valore approssimate $v_0, v_1, v_2, \ldots \epsilon \mathbb{R}$ dove v_0 è scelta arbitrariamente. Ogni approssimazione successiva (v_1, v_2, ecc.) si ottiene come iterazione dell'Equazione 3.1, ossia dall'equazione di Bellman per $v_\pi(s)$, da cui si ottiene la seguente espressione:

$$v_{k+1}(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) | S_t = s]$$
$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right] \quad (3.2)$$

per ogni $s \epsilon S$. L'obiettivo di questo processo iterativo è approssimare la funzione valore ripetutamente $(k, k+1, \ldots)$ finché la sequenza v_k converge in v_π per $k \rightarrow \infty$, cioè man mano che si itera il valore v_k si avvicina sempre più al suo valore vero v_π. Questo processo è chiamato valutazione iterativa della *policy* (dall'inglese *iterative policy evalutation*). Per generare ogni approssimazione successiva della funzione valore, cioè v_{k+1}, la valutazione iterativa della *policy* applica la stessa operazione a ciascuno stato s: il valore di $v_{k+1}(s)$ viene calcolato considerando la ricompensa attesa (cioè r) allo stato successivo s' come conseguenza dell'azione a compiuta in s, sommata alla stima del valore degli stati successivi, cioè $v_k(s')$, pesati per le probabilità (come descritto nell'Equazione 3.2). Questo meccanismo viene chiamato aggiornamento atteso (da *expected*

update in Barto e Sutton, 2018). Ciascuna iterazione aggiorna il valore di ogni stato una volta per produrre la nuova funzione valore v_{k+1} approssimata. Man mano che l'iterazione prosegue, il valore si avvicina sempre di più al valore vero v_π, ché è l'obiettivo di questa prima fase di valutazione. Esistono diversi tipi di aggiornamento atteso, a seconda che venga aggiornato uno stato (come nell'Equazione 3.2) o una coppia stato-azione (se si usa una funzione valore stato-azione). Tutti gli aggiornamenti effettuati negli algoritmi di programmazione dinamica sono chiamati "aggiornamento atteso" perché si basano su un'aspettativa su tutti i possibili stati successivi (nella formula $v_k(s'')$), piuttosto che su un loro campione.

3.2.2 Seconda fase: il miglioramento della *policy*

Una volta ottenuto il valore di v_π, si sa quanto valore abbia mantenere l'attuale *policy*, ma se invece si cambiasse sarebbe meglio o peggio? Con la fase di miglioramento della *policy* si vuole capire se sia meglio mantenere la *policy* arbitraria della prima fase ($\pi(s)$) oppure cambiarla selezionando un'altra azione $a \neq \pi(s)$.

Per rispondere a questa domanda si può calcolare la funzione valore usando la nuova azione a, seguendo la *policy* arbitraria usata nella prima fase, cioè $\pi(s)$. Il valore in questo modo è dato dalla funzione valore stato-azione:

$$\begin{aligned} q_\pi(s,a) &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = a] \\ &= \sum_{s',r} p(s', r|s, a)\,[r + \gamma v_\pi(s')]. \end{aligned} \tag{3.3}$$

Se il valore di $q_\pi(s,a)$ che si ottiene è maggiore del valore $v_\pi(s)$ di partenza (cioè il valore ottenuto dalla prima fase) significa che l'azione a è una scelta migliore rispetto alla *policy* precedente. Al contrario, se il valore ottenuto con l'Equazione 3.3 è minore del valore $v_\pi(s)$ significa che non c'è convenienza a selezionare la nuova azione a. Quindi, dato che π e π' sono uguali tranne per

un'azione $a \neq \pi(s)$ associata allo stato s, la *policy* modificata è migliore di quella arbitraria perché porta ad un maggiore valore allo stato s.

Questo è un risultato generale chiamato teorema di miglioramento della *policy* (Barto e Sutton, 2018). Secondo il teorema, siano π e π' una coppia di *policies* tale che, per ogni $s \epsilon S$,

$$q_\pi\big(s, \pi'(s)\big) \geq v_\pi(s) \tag{3.4}$$

cioè il valore dello stato s selezionando la nuova azione $\pi'(s)$ è maggiore del valore precedente con la *policy* iniziale. Allora, in generale, la policy π' porta risultati attesi migliori o pari alla policy π in ogni stato $s \epsilon S$:

$$v_{\pi'}(s) \geq v_\pi(s) \tag{3.5}$$

Inoltre, se nell'Equazione 3.4 la disuguaglianza è strettamente maggiore per ogni stato $s \epsilon S$ allora nell'Equazione 3.5 ci deve essere il segno di disuguaglianza stretta per almeno uno stato.

Con l'Equazione 3.3 si è visto come si possa facilmente valutare un cambiamento della *policy* modificando l'azione in uno stato s. Un'estensione naturale considera i cambiamenti in tutti gli stati per tutte le azioni possibili, selezionando ad ogni stato l'azione che risulta migliore sulla base di $q_\pi(s, a)$. In termini matematici, la nuova *greedy-policy*, π', è data dall'espressione

$$\begin{aligned} \pi'(s) &= arg \max_a q_\pi(s, a) \\ &= arg \max_a \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = a] \\ &= arg \max_a \sum_{s', r} p(s', r | s, a)[r + \gamma v_\pi(s')] \end{aligned} \tag{3.6}$$

dove $argmax_a$ indica il valore di a per cui viene massimizzata l'espressione che lo segue, cioè la funzione valore stato-azione. Questa *policy* seleziona l'azione che sembra migliore a breve termine, cioè nell'ottica del periodo successivo s'. Per costruzione, la *greedy-policy* soddisfa le condizioni del teorema di miglioramento

della *policy* (Equazione 3.4), quindi è migliore o "buona" quanto la *policy* iniziale. Dunque, la fase di miglioramento migliora una *policy* iniziale rendendola *greedy* rispetto alla funzione valore data (che è il risultato della fase di valutazione della *policy*), quindi l'obiettivo è quello di trovare la politica che ti dà la maggior ricompensa che puoi ricevere scelta l'azione migliore. Questa fase viene anche chiamata problema di controllo.

Si supponga una *greedy-policy*, π', uguale alla *policy* iniziale π. Allora $v_\pi = v_{\pi'}$ e, dall'Equazione 3.6, si ha che per ogni $s \epsilon S$:

$$\begin{aligned} v_{\pi'}(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_{\pi'}(S_{t+1}) \,|S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_{\pi'}(s')]. \end{aligned} \tag{3.7}$$

L'espressione è esattamente la stessa dell'equazione di ottimalità di Bellman (Equazione 2.11), dove $v_{\pi'} = v_*$. Di conseguenza, se la *policy* $v_{\pi'}$ è uguale alla precedente v_π, significa che non c'è più margine di miglioramento e quindi la *policy* π' è ottimale come per l'Equazione 2.11. La fase di miglioramento della *policy* permette di ottenere una *policy* migliore, tranne quando la *policy* iniziale è già ottimale.

3.2.3 Due algoritmi di programmazione dinamica: *Policy Iteration* e *Value Iteration*

Una volta che la *policy* π è stata migliorata utilizzando v_π e ottenendo π', si può calcolare la funzione valore $v_{\pi'}$, ovvero la funzione valore relativa alla *policy* migliorata, per implementarla a sua volta e ottenere un ulteriore miglioramento che generi π''. Si ottiene così una sequenza di miglioramento crescente come segue:

$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi_* \xrightarrow{E} v_*$$

dove $\xrightarrow{E}$ indica l'applicazione della fase di valutazione della *policy* ("E" da *evaluation*), mentre $\xrightarrow{I}$ la fase di miglioramento ("I" da *improvement*). In questo modo si garantisce un miglioramento della *policy* precedente, a meno che non sia già ottimale. Dato che un MDP finito ha un numero finito di *policies*, il processo converge alla *policy* ottimale e alla funzione valore ottimale con un numero finito di iterazioni. Questo processo è uno degli algoritmi di programmazione dinamica e prende il nome di *Policy Iteration*.

Uno svantaggio della *Policy Iteration* è che la fase di valutazione della *policy* comporta un calcolo iterativo che può essere indefinitamente lungo in attesa della convergenza esatta a v_π.

Un altro algoritmo appartenente alla programmazione dinamica è chiamato *Value Iteration* che compie una ricerca della funzione del valore ottimale e, solo quando è stata determinata v_*, un'estrazione della *policy* π_*. Non si alternano le due operazioni in questo algoritmo, perché una volta che la funzione valore è ottimale, anche la *policy* associata ad essa dovrebbe essere ottimale. Formalmente la ricerca del valore ottimo consiste nella seguente equazione:

$$\begin{aligned} v_{k+1}(s) &= \max_a E[R_{t+1} + \gamma v_k(S_{t+1}) \,|S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_k(s')] \end{aligned} \tag{3.8}$$

per ogni $s \epsilon S$. Con questa equazione, si aggiorna iterativamente il valore di ogni stato per ottenerne il valore successivo, costruendo per ogni stato una sequenza del tipo:

$$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow \cdots \rightarrow v_*$$

È possibile dimostrare che la sequenza v_k converge a v_* assumendo le stesse condizioni che garantiscono l'esistenza di v_*. Si noti che l'algoritmo di *Value Iteration* si ottiene semplicemente trasformando l'equazione di ottimalità di Bellman per v_* (Equazione 2.11) in aggiornamento atteso (come per la fase di valutazione della *policy*).

La caratteristica che contraddistingue l'algoritmo di *Value Iteration* è l'operazione di massimizzazione che si compie nella fase di valutazione della funzione valore, che rende diversa l'Equazione 3.8 dalla più generica equazione di valutazione della *policy* introdotta con l'Equazione 3.2. Questo operatore fa sì che si compia una massimizzazione su tutte le azioni a, come se si realizzasse in modo intrinseco una fase di miglioramento ad ogni fase di valutazione della funzione valore.

Per puntualizzarne le differenze, si vedano le figure 3.3 e 3.4 che mostrano, rispettivamente, la struttura dell'algoritmo di *Policy Iteration* e *Value Iteration*. Il procedimento del primo algoritmo al termine del passaggio 3 (miglioramento della *policy*) dice che se la *policy* è stabile, cioè se $\pi = \pi'$, allora si ferma il processo perché ha raggiunto il valore ottimale, altrimenti finché la *policy* non è stabile si ripetono alternando i passaggi 2 e 3. Dal riquadro che descrive l'algoritmo *Value Iteration* si nota come non ci siano questi passaggi. Piuttosto, si ripetono le iterazioni per ottenere la funzione valore e, solo dopo che si ottengono variazioni molto piccole tra un'iterazione e l'altra (dunque è ottimale), si determina la sua *policy* associata. Si noti infine l'operatore di massimizzazione nelle iterazioni di $V(s)$ che esprime la differenza chiave tra i due algoritmi.

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
      $\Delta \leftarrow 0$
      Loop for each $s \in \mathcal{S}$:
         $v \leftarrow V(s)$
         $V(s) \leftarrow \sum_{s',r} p(s', r|s, \pi(s))\left[r + \gamma V(s')\right]$
         $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
      old-action $\leftarrow \pi(s)$
      $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r|s, a)\left[r + \gamma V(s')\right]$
      If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Figura 3.3: Meccanismo dell'algoritmo Policy Iteration. Fonte: Barto e Sutton, 2018.

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r|s, a)\left[r + \gamma V(s')\right]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \text{argmax}_a \sum_{s',r} p(s', r|s, a)\left[r + \gamma V(s')\right]$

Figura 3.4: Meccanismo dell'algoritmo Value Iteration. Fonte: Barto e Sutton, 2018.

3.3 Metodo Monte Carlo

Il metodo Monte Carlo stima le funzioni valore e le *policies* ottimali attraverso l'esperienza, ossia per mezzo di campioni. A differenza della programmazione dinamica, non si assume un modello noto (cioè probabilità di transizione e *rewards*) poiché il metodo Monte Carlo utilizza un campione di stati, azioni e *rewards* che ottiene interagendo con l'ambiente. I campioni sono costituiti da realizzazioni possibili del fenomeno che si sta esaminando che sono generate dal processo per mezzo di simulazioni dell'esperienza dell'agente con l'ambiente, così da essere utilizzati per calcolare numericamente le funzioni di valore al posto del modello. Le funzioni valore e le *policies* corrispondenti interagiscono per raggiungere l'ottimalità essenzialmente allo stesso modo della programmazione dinamica, cioè alternando le fasi del principio GPI.

Una caratteristica dei metodi Monte Carlo, come viene spiegato tra poco in dettaglio, è che il processo si basa sulla media dei *rewards* dei campioni. Per garantire dei *rewards* ben definiti, i processi sono strutturati per episodi e non per periodi ($t = 0,1, \ldots$): questo significa che il processo è finito e di lunghezza T (cioè $t = 0, \ldots, T$) e, mentre prima le valutazioni venivano compiute ad ogni t (*step-by-step*), ora le valutazioni sono compiute ad episodi, cioè sono una volta

che si raggiunge il termine T. Per esempio, se il gioco del Tic-Tac-Toe fosse risolto con la programmazione dinamica il processo si svolgerebbe tra una mossa e l'altra, se invece usasse il metodo Monte Carlo un episodio sarebbe una partita intera al termine della quale si calcolano e aggiornano le stime. Quindi, solo al completamento dell'episodio vengono modificate le stime delle funzioni valore e le *policies*.

Relativamente alla fase di valutazione della *policy*, la logica sottostante ai metodi Monte Carlo è stimare il valore di uno stato sulla base dell'esperienza, attraverso la media dei *rewards* osservati dopo che si è visitato più volte quello stato. In questo modo si ottiene il campione di *rewards*, dei quali la media converge al valore atteso durante le iterazioni. Per comprendere meglio, si veda il confronto tra il diagramma della programmazione dinamica e quello del metodo Monte Carlo, rispettivamente illustrati a sinistra e a destra nella Figura 3.2. Il metodo Monte Carlo inizia dallo stato di cui si vuole aggiornare la stima della funzione valore, rappresentato dal nodo più alto. A partire da questo vengono rappresentate tutte le transizioni che sono compiute fino al termine dell'episodio, i cui stati e i *rewards* contribuiscono all'aggiornamento del valore del nodo iniziale. Si noti che, a differenza del diagramma della programmazione dinamica dove sono rappresentate tutte le possibili transizioni da uno stato di partenza a quello successivo, nel metodo Monte Carlo si illustrano solo quelle che vengono visitate durante l'episodio. In più, a sinistra si rappresenta una sola transizione da stato a stato, mentre a destra tutte le transizioni fino al termine dell'episodio. Queste sono

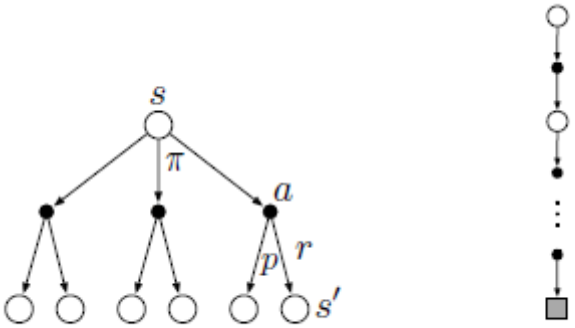

Figura 3.2: Diagrammi di backup per la programmazione dinamica (sinistra) e il metodo Monte Carlo (destra). Fonte: Barto e Sutton (2017).

le differenze principali tra i due metodi.

Una caratteristica importante dei metodi Monte Carlo è che le stime per ciascuno stato sono indipendenti. La stima per uno stato non si basa sulla stima di un altro stato, come nel caso della programmazione dinamica. Inoltre, le risorse necessarie per la stima della funzione valore di uno stato non dipendono dal numero totale degli stati, perché si basano sui campioni relativi allo stato in esame raccolti nei vari episodi. Questo rende il metodo Monte Carlo conveniente quando si richiede il valore di uno stato o alcuni, poiché possono generarsi episodi a partire dallo stato di interesse calcolando la media dei rendimenti che si ottengono nel processo, ignorando gli altri.

Dato che un modello per l'ambiente non è disponibile, allora è necessario stimare la funzione valore stato-azione piuttosto che la funzione valore dello stato. In presenza di un modello è sufficiente calcolare la funzione valore $v(s)$ per ottenere una *policy*, senza un modello invece i valori degli stati non sono sufficienti, ma è necessario esplicitare il valore di ogni possibile azione per determinare la *policy*. Nella fase di valutazione della *policy* l'obiettivo diventa quindi la stima di $q_\pi(s, a)$ partendo dallo stato s, eseguendo l'azione a e data la *policy* π. Più in generale l'obiettivo del processo diventa la stima di q_* e π_*.

Il problema che si presenta con questa differenza è che calcolando $q_\pi(s, a)$ alcune coppie stato-azione potrebbero non essere mai visitate, poiché la *policy* π fa sì che l'azione associata ad uno stato sarà sempre la stessa e si otterrà la stima dei *rewards* solo per una delle azioni possibili. In questo modo non è possibile migliorare le stime delle altre azioni con l'esperienza. Dato che lo scopo di apprendere i valori delle azioni è di aiutare a scegliere tra le azioni disponibili in ogni stato, questo problema risulta limitante. Per confrontare le alternative si deve stimare il valore di tutte le azioni di uno stato, non solo considerare quello che attualmente si preferisce. Questo problema viene chiamato problema del mantenimento dell'esplorazione. Affinché la fase di valutazione della *policy* funzioni anche con la funzione di valore stato-azione, si deve garantire una continua esplorazione, cioè provare delle azioni che prima non ha sperimentato così da valutarne i risultati. Un modo per farlo è quello di specificare che negli episodi ogni coppia stato-azione ha probabilità diversa da zero di essere

selezionata come nodo iniziale. Questo metodo si chiama avvio esplorativo e garantisce che tutte le coppie stato-azione saranno visitate un numero infinito di volte quando il numero degli episodi tende ad infinito.

Il meccanismo generale che guida il processo Monte Carlo all'approssimazione di una *policy* ottimale funziona seguendo il principio di GPI come per la *Policy Iteration* della programmazione dinamica, con l'unica differenza di utilizzare la funzione valore $q_\pi(s,a)$ piuttosto che $v_\pi(s)$. Con il GPI la funzione valore viene iterativamente aggiornata per approssimare meglio la funzione valore per la *policy* corrente, mentre la *policy* viene ripetutamente migliorata rispetto alla funzione valore corrente, come suggerito dalla Figura 3.3.

La valutazione della *policy* viene eseguita come nella programmazione dinamica: si considerano molti episodi nei quali la funzione valore stato-azione approssimata si avvicina asintoticamente al suo valore vero. Si supponga che si considerino un numero infinito di episodi e che siano generati con l'avvio esplorativo. Con queste ipotesi, i metodi Monte Carlo calcoleranno esattamente ogni q_{π_k} per la *policy* arbitraria π_k. La fase di miglioramento della *policy* si realizzerà con una *greedy-policy* rispetto alla funzione valore in uso: per qualsiasi funzione q, la corrispondente *greedy policy* è quella che, per ogni $s \in S$, sceglie un'azione con il massimo valore associato alla coppia stato-azione, cioè:

$$\pi(s) = \arg\max_a q(s,a). \tag{3.9}$$

Alternando quindi le due fasi, valutazione e miglioramento della *policy*, si ottiene

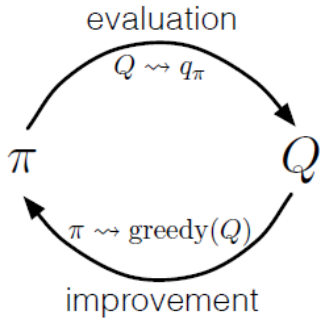

Figura 3.3: principio di GPI per il metodo Monte Carlo

una sequenza come la seguente, partendo da un'arbitraria policy π_0:

$$\pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi_* \xrightarrow{E} q_*$$

Come discusso nel paragrafo precedente, il teorema di miglioramento della *policy* assicura che ogni π_{k+1} è migliore di π_k o al più uguale (nel qual caso entrambe sono *policies* ottimali). Questo a sua volta fa sì che l'intero processo converga[25] alla *policy* ottimale e alla funzione valore ottimale. In questo modo i metodi Monte Carlo possono essere utilizzati per trovare *policies* ottimali dati solo episodi di esempio e senza nessuna conoscenza delle dinamiche dell'ambiente.

3.4 Apprendimento per differenze temporali

Quando si fa riferimento al RL, risulta centrale il ruolo dei metodi di apprendimento per differenze temporali, la classe degli algoritmi più studiati e utilizzati. Questi processi sono una combinazione di caratteristiche della programmazione dinamica e dei metodi Monte Carlo. Infatti, come nei metodi Monte Carlo, non richiedono un modello noto per l'ambiente, bensì possono apprendere direttamente dall'esperienza. Come nella programmazione dinamica aggiornano le stime sulla base di altre stime, senza dover attendere un risultato finale.

Per capire meglio cosa si intende per "stime sulla base di altre stime" si prenda d'esempio una persona che deve decidere a che ora far arrivare a casa gli ospiti a cena. Prima di cucinare deve passare al supermercato, dal macellaio e in enoteca, in questo ordine. Stima il tempo che impiegherà per le commissioni e prevede di riuscire a passare dal macellaio e in enoteca in 10 minuti, mentre, data l'ora di punta, stima di impiegare 30 minuti al supermercato. Sulla base di queste

[25] Si sono fatte due ipotesi improbabili per ottenere la convergenza: la prima stabilisce che gli episodi abbiano un avvio esplorativo e la seconda richiede che la valutazione della *policy* sia compiuta con un numero infinito di episodi. Tuttavia, per ottenere un algoritmo operativo si dovrebbero rimuovere entrambi i presupposti. Questo argomento non è utile ai fini dell'elaborato, per cui non verrà esposto. Per approfondimenti a riguardo si veda Barto e Sutton, 2018.

previsioni, dice agli ospiti di arrivare alle 18.00. Una volta usciti dal supermercato realizza che gli sono bastati 10 minuti per fare la spesa, per cui aggiusta le stime prevedendo di arrivare a casa con 20 minuti di anticipo. Tuttavia, una volta lasciato il macellaio per raggiungere l'enoteca si blocca nel traffico ed impiega 30 minuti in più per arrivare. Alla fine, arriva a casa con 10 minuti di ritardo rispetto alla previsione iniziale (Wiering, 2005). Questo esempio sottolinea che si può aggiornare la stima iniziale ogni volta che subentra una nuova informazione tra un passaggio e l'altro. Quindi ogni aggiornamento sulla previsione iniziale si basa sull'esperienza che si acquisisce lungo il percorso. Questo è il principio generale dell'apprendimento per differenze temporali: non si deve attendere fino alla fine per aggiornare le previsioni lungo il processo. In più, come anticipato, questo meccanismo ha il vantaggio di non aver bisogno di una dinamica di ambiente poiché la acquisisce lungo il processo.

Per capire meglio, si veda il seguente schema che rappresenta la logica principale sottostante ai metodi in esame:

$$Nuova\ stima \leftarrow Vecchia\ stima + \frac{Tasso\ di}{apprendimento}\,[Target - Vecchia\ stima] \tag{3.10}$$

Questa logica rappresenta l'aggiornamento che viene compiuto quando si approssima la funzione valore al suo valore vero. Come già visto, a partire dalla "vecchia" stima, intesa come la precedente, si aggiorna questo valore con una frazione di errore (misurato come differenza tra lobiettivo, "target", e la "vecchia" stima), la quale dipende dalla dimensione del tasso di apprendimento del processo.

Applicando questo schema al metodo Monte Carlo, possiamo definire l'aggiornamento come segue:

$$V_{k+1}(S_t) \leftarrow V_k(S_t) + \alpha_k[G_t - V_k(S_t)] \tag{3.11}$$

dove $V(S_t)$ è la stima della funzione valore $v_\pi(s)$ allo stato S_t, G_t è il *reward* ricevuto e α_t è la costante *step-size parameter* chiamata tasso di apprendimento (o

learning rate), che rappresenta la velocità con cui il modello apprende e determina la dimensione del passo dell'iterazione, cioè quanto pesa l'errore in quell'iterazione.
In (3.11) l'aggiornamento $(k+1)$-esimo si ottiene dalla somma della k-esima stima e della quantità nelle parentesi, che rappresenta la distanza della k-esima stima da un target, i *rewards* ricevuti G_t, che costituiscono una sorta di livello desiderabile.

Passando oltre, anche l'apprendimento per differenze temporali aggiorna le stime di v_π attraverso l'esperienza ma, al contrario di Monte Carlo, non attende sia dato il G_t per usarlo come target di $V(S_t)$, come invece si poteva vedere dall'Equazione 3.11. Con il metodo per differenze temporali l'apprendimento viene fatto ad ogni passo t, non direttamente al termine dell'episodio. Questo tipo di approccio viene chiamato incrementale e il suo schema di aggiornamento più semplice e immediato è descritto come

$$V_{k+1}(S_t) \leftarrow V_k(S_t) + \alpha_k[R_{t+1} + \gamma V_k(S_{t+1}) - V_k(S_t)]. \tag{3.12}$$

Quindi la differenza tra i due aggiornamenti della funzione valore risiede nel target: nel metodo Monte Carlo il target è la funzione dei rendimenti realizzati G_t al termine degli episodi, invece nei metodi per differenze temporali l'aggiornamento si basa sulla lontananza della stima $V_k(S_t)$ dalla componente target $R_{t+1} + \gamma V(S_{t+1})$. Questo metodo è chiamato $TD(0)$ o *one-step* TD[26].

Per sottolineare la differenza tra tutti i tre metodi, si veda la definizione della funzione valore come definita dall'Equazione 2.10:

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] \tag{3.13}$$
$$= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s]$$
$$= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s]. \tag{3.14}$$

I metodi Monte Carlo utilizzano una stima del target come formulata nella (3.13), mentre i metodi di programmazione dinamica utilizzano una stima come

[26] Il *one-step* TD è un caso speciale della più generale classe TD(λ), che non viene presentata nell'elaborato.

formulata nella (3.14). Dato che il valore atteso $\mathbb{E}_\pi(G_t)$ in (3.13) non è noto, il target G_t utilizzato dai metodi Monte Carlo in (3.11) è una stima. Allo stesso modo, nella programmazione dinamica il target è rappresentato dalla stima $V_k(S_{t+1})$ perché il valore vero $v_\pi(S_{t+1})$ non è noto. Il target nei metodi per differenza temporale (cioè la quantità $R_{t+1} + \gamma V(S_{t+1})$) è una stima per entrambi i motivi: campiona i valori R_{t+1} (come Monte Carlo) e utilizza la stima corrente $V_k(S_{t+1})$ al posto del valore vero (come nella programmazione dinamica). Questo è il motivo per cui i metodi per differenze temporali combinano elementi della programmazione dinamica e di Monte Carlo.

Vi sono alcuni approfondimenti da fare in merito al parametro α_t, cioè al tasso di apprendimento poiché la sua dimensione determina la convergenza o meno dell'iterazione della funzione valore. In particolare quando α_t è variabile al tempo t le condizioni di convergenza richieste secondo le teoria di approssimazione stocastica sono:

$$\sum_{n=1}^{\infty} a_n(a) = \infty \; e \; \sum_{n=1}^{\infty} \alpha_n^2(a) < \infty \, . \tag{3.15}$$

La prima condizione è richiesta per delle condizioni di instabilità che possono presentarsi inizialmente, la seconda condizione garantisce che alla fine si arrivi al punto in cui i passaggi diventino abbastanza piccoli da garantire la convergenza. Quando si utilizza invece un α_t costante la seconda condizione non è soddisfatta, indicando che le stime non convergono mai completamente ma continuano a variare in risposta ai valori dei rendimenti più recenti. Questa condizione è tipica in un ambiente non stazionario com'è quello dei mercati finanziari, nel quale conviene non appesantire il processo di ricerca con osservazioni dei rendimenti troppo lontane nel tempo, ma piuttosto è meglio focalizzarsi sulle osservazioni più recenti. Se il parametro α_t viene settato sufficientemente piccolo, comunque la stima della funzione valore convergerà al valore ottimo, sia in condizioni di stazionarietà, che di non stazionarietà. È dunque necessario trovare il valore migliore per il parametro α_t, abbastanza piccolo per ottenere la convergenza, ma non troppo per non renderla troppo lenta (Barto e Sutton, 2018).

Infine, si noti che la quantità all'interno della parantesi dell'Equazione (3.12) rappresenta una sorta di errore, poiché misura la differenza tra un valore stimato di S_t, cioè $V_k(S_t)$, e la sua miglior stima $R_{t+1} + \gamma V(S_{t+1})$, cioè il target. Questa quantità, indicata con δ_t, è chiamata errore TD e si trova in forme diverse nei vari metodi di RL, a seconda dell'algoritmo che si sta applicando:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t). \tag{3.16}$$

Si noti che l'errore TD è l'errore nella stima compiuta in t. Dato che l'errore TD dipende dallo stato e dai *rewards* in $t + 1$, non è disponibile fino allo stato successivo. Ciò significa che δ_t è l'errore nella stima $V(S_t)$ disponibile al tempo $t + 1$.

Come già detto più volte, un vantaggio dei metodi di apprendimento per differenze temporali rispetto alla programmazione dinamica è che non richiedono la conoscenza di un modello per l'ambiente. Invece, il vantaggio più ovvio rispetto ai metodi Monte Carlo, è che l'apprendimento avviene man mano che si ricevono i dati (chiamato apprendimento *on-line*). In altre parole, mentre con i metodi Monte Carlo si attende fino alla fine di un episodio perché solo allora si conosce il *reward*, con i metodi per differenze temporali è necessario attendere solo un passaggio, cioè da t a $t + 1$.

3.4.1 Gli algoritmi: SARSA, Q-*Learning*, *Greedy*-GQ

Gli algoritmi che si presentano di seguito appartengono alla classe dei metodi per differenze temporali sono il metodo SARSA, Q-*Learning*, *Greedy*-GQ.
Il primo algoritmo, SARSA, viene rappresentato con il seguente schema incrementale di aggiornamento

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha_t[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \tag{3.17}$$

dove l'aggiornamento si realizza come aggiustamento della stima per mezzo di un errore, cioè la quantità tra partesi che rappresenta la differenza della stima

$Q(S_t, A_{t+1})$ dalla componente target $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$ moltiplicata per il parametro *step-size*. Il meccanismo dell'algoritmo è tale che una transizione stato-azione da t a $t+1$ genera la sequenza di elementi $S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}$ da cui l'algoritmo prende il nome. In SARSA le azioni sono scelte in ogni stato t utilizzando una *policy ε-greedy* rispetto a Q, cioè:

$$\begin{cases} A_t \in arg \max\limits_{a \in A(S_t)} Q(S_t, a) & \text{con probabilità } 1 - \varepsilon_t \\ A(S_t) & \text{con probabilità } \varepsilon_t \end{cases}$$

con $0 < \varepsilon_t \ll 1$. In altre parole, con probabilità $1 - \varepsilon_t$ l'azione selezionata è quella a cui è associata la funzione valore stato-azione con valore maggiore. Invece, con probabilità ε_t viene scelta un'azione casuale. Con questo parametro ε_t si permette quindi all'algoritmo di ottenere un corretto bilanciamento del trade-off tra *exploration* ed *exploitation*[27] (Corazza *et. al*, 2019).

Il secondo algoritmo, il Q-*Learning*, è l'algoritmo più diffuso e conosciuto dei metodi per differenze temporali. Il suo schema incrementale di aggiornamento è

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha_t \left[R_{t+1} + \gamma \max_{a \in A(S_{t+1})} Q(S_{t+1}, a) - Q(S_t, A_t) \right]. \qquad (3.18)$$

In entrambi gli algoritmi (3.17) e (3.18) il tasso di apprendimento può vedersi come un tasso che indica quanto le nuove informazioni apprese incidono su quelle vecchie; se infatti il tasso fosse 0 non ci sarebbe apprendimento perché la componente di errore sarebbe pari a 0, al contrario se fosse 1 significa che l'agente dà grande importanza solo a quanto appreso. Il fattore di sconto γ invece attribuisce più o meno importanza ai valori futuri.

[27] Come introdotto nel Capitolo 2, il *trade-off* tra *exploration* ed *exploitation* è il bilanciamento tra esplorare nuove azioni e sfruttare quello che si è sperimentato. In generale, per ottenere una ricompensa positiva come conseguenza di un'azione, un agente di Reinforcement Learning preferisce azioni che ha già provato nel passato e che ha scoperto essere proficue. Tuttavia, l'agente riceve solo il segnale numerico come conseguenza, non sa qual è l'azione migliore. Per scoprirlo deve provare altre azioni possibili che non ha mai compiuto in passato (*exploration*), così da capire quali restituiscono ricompense positive migliori (*exploitation*). In altre parole, l'agente deve sfruttare quello che già ha sperimentato per ottenere ricompense positive, ma deve anche tentare nuove azioni per fare scelte migliori nel futuro.

Dato che in (3.18) le azioni sono nuovamente selezionate in base a una *policy ε-greedy* rispetto a $Q(S_{t+1}, a)$, la differenza tra Q-*learning* e SARSA è che il primo è un algoritmo di controllo *off-policy*, il che significa che il fattore di correzione viene determinato utilizzando un'azione eventualmente diversa da quella effettivamente scelta (Corazza *et al.*, 2019). Infatti, l'algoritmo Q-*Learning* usa due *policies* diverse: una viene utilizzata per stimare le funzioni di valore, un'altra viene utilizzata per controllare il processo di miglioramento. In questo modo si semplifica l'analisi e permette di raggiungere la convergenza più velocemente. SARSA è invece un algoritmo di controllo *on-policy*, cioè le stime di $Q(S_t, A_t)$ vengono aggiornate utilizzando un fattore di correzione fornito da azioni selezionate con la *policy* corrente, cioè la stessa utilizzata per la stima delle funzioni valore.

È possibile dimostrare che entrambi gli algoritmi convergono alla funzione valore stato-azione ottimale, a condizione che tutte le coppie stato-azione che siano visitate un numero infinito di volte e siano soddisfatte alcune condizioni sui parametri α e ε_t (Singh *et al.*, 2000). Il primo requisito è difficile da garantire anche in maniera approssimativa in quanto generalmente si ha a che fare con un elevato numero di stati che comporta un problema di risorse computazionali necessarie. Al fine di ovviare a questo problema, esiste un altro approccio che consiste nella generalizzazione attraverso opportune approssimazioni delle funzioni valore stato-azione. Queste vengono rappresentate con la forma di funzioni parametrizzate con un vettore di pesi $\boldsymbol{w} \in \mathbb{R}^d$ di dimensione d, da cui la stima della funzione valore stato-azione diventa $Q(s, a, \boldsymbol{w})$. L'obiettivo è quello di ottenere l'approssimazione della funzione valore più corretta possibile, che in altri termini può vedersi come la minore differenza possibile tra la stima della funzione valore e il suo valore vero. Quindi, si tratta di stimare il vettore ottimale dei parametri, $\boldsymbol{w}_*$, tale che minimizzi l'errore di stima. Nella maggior parte dei modelli di apprendimento viene utilizzata la minimizzazione dell'errore quadratico medio[28], indicato con $\overline{VE}$:

[28] Non è del tutto chiaro se l'errore quadratico medio sia la misura di performance migliore per il Reinforcement Learning. Si ricorda che lo scopo ultimo – il motivo per cui si cerca la funzione valore – è quello di trovare la miglior *policy*. La migliore funzione valore per questo scopo non è detta sia la stessa che minimizza nel modo migliore l'errore quadratico medio. Tuttavia, non è

$$\overline{VE}(\boldsymbol{w}) = \sum_{s\epsilon S} \mu(s) \sum_{a\epsilon A} \pi(s,a)\,[q_\pi(s,a) - Q(s,a,\boldsymbol{w})]^2, \tag{3.19}$$

dove $\mu(s)$ è una qualche distribuzione di probabilità di $\mathcal{S}$ (con $\mathcal{S}$ finito), quindi $\mu(s) \geq 0$ e $\sum_S \mu(s) = 1$, che rappresenta l'importanza[29] dell'errore ad ogni stato s. Il vettore $\boldsymbol{w}$ diventa quindi il vettore incognita in questo problema di ottimizzazione.

L'obiettivo ideale in termini di $\overline{VE}$ è quello di ottenere un ottimo globale, cioè un vettore di pesi $\boldsymbol{w}_*$ tale che $\overline{VE}(\boldsymbol{w}_*) \leq \overline{VE}(\boldsymbol{w})$ per ogni $\boldsymbol{w}$. Questo obiettivo è talvolta possibile da raggiungere per funzioni di approssimazione semplici come quelle lineari, mentre è più difficile da raggiungere per funzioni di approssimazione complesse. In alternativa, l'obiettivo può diventare quello di convergere ad un ottimale locale, cioè un vettore di pesi $\boldsymbol{w}$ tale che $\overline{VE}(\boldsymbol{w}_*) \leq \overline{VE}(\boldsymbol{w})$ per ogni valore $\boldsymbol{w}$ in un intorno di $\boldsymbol{w}_*$. Sebbene questa garanzia sia solo parzialmente rassicurante, è in genere la soluzione migliore per funzioni di approssimazione non lineari e spesso è sufficiente.

Tra tutti i metodi di approssimazione delle funzioni i più utilizzati sono i metodi di discesa stocastica del gradiente (dall'inglese *Stochastic Gradient Descent*, abbreviato con SGD), particolarmente adatti al RL. In questi metodi il vettore di pesi è un vettore colonna $\boldsymbol{w}^T = (w_1, w_2, \ldots, w_d)^T$, mentre la stima della funzione valore parametrizzata $Q(s,a,\boldsymbol{w})$ è una funzione differenziabile per $\boldsymbol{w}$ e per ogni $s\epsilon S$. Il valore del vettore $\boldsymbol{w}$ viene aggiornato ad ogni t per cui è più utile utilizzare la notazione temporale $\boldsymbol{w}_t$ ad ogni step. La discesa stocastica del gradiente è un metodo iterativo che ad ogni periodo t aggiorna il vettore di pesi $\boldsymbol{w}$ di una piccola quantità nella direzione che riduce maggiormente l'errore in quel periodo, cioè

---

ancora chiaro quale potrebbe essere un obiettivo alternativo più utile per la previsione del valore (Barto e Sutton, 2018).

[29] Si rende necessario usare una distribuzione di probabilità per i possibili stati perché quando si aggiorna il valore di uno stato, questo influisce di riflesso anche sul valore di altri stati, modificandoli. Come conseguenza significa che rendere più accurata una stima potrebbe rendere meno accurate le altre. Utilizzando una distribuzione di probabilità invece si può dare attenzione maggiore agli stati che richiedono più precisione nella stima. (Barto e Sutton, 2018).

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha_t[q_\pi(s,a) - Q(S_t, A_t, \boldsymbol{w}_t)]\nabla_{\boldsymbol{w}_t} Q(S_t, A_t, \boldsymbol{w}_t), \tag{3.20}$$

dove α è il parametro step-size e $\nabla_{\boldsymbol{w}} Q(S_t, A_t, \boldsymbol{w}_t)$ rappresenta il vettore di derivate parziali rispetto alle componenti del vettore di pesi $\boldsymbol{w}_t$, cioè

$$\nabla_{\boldsymbol{w}_t Q(S_t, A_t, \boldsymbol{w}_t)} = \left(\frac{dQ(S_t, A_t, \boldsymbol{w}_t)}{d\boldsymbol{w}_{t_1}}, \frac{dQ(S_t, A_t, \boldsymbol{w}_t)}{d\boldsymbol{w}_{t_1}}, \dots, \frac{dQ(S_t, A_t, \boldsymbol{w}_t)}{d\boldsymbol{w}_{t_1}}\right)' \tag{3.21}$$

chiamato anche gradiente di Q rispetto a $\boldsymbol{w}$.

Il valore iniziale di $q_\pi(s,a)$ dell'Equazione 3.20 non è noto, ma lo si può sostituire con una stima target U_t che approssima il valore vero. L'Equazione 3.20 diventa quindi:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha[U_t - Q(S_t, A_t, \boldsymbol{w}_t)]\nabla_{\boldsymbol{w}_t} Q(S_t, A_t, \boldsymbol{w}_t). \tag{3.22}$$

Se U_t è uno stimatore non distorto, cioè se $\mathbb{E}[U_t | S_t = s] = v_\pi(S_t)$ per ogni t, allora $\boldsymbol{w}_t$ converge ad un ottimo locale se valgono le già note condizioni di α.

Il caso speciale più importante è l'utilizzo di una funzione di approssimazione lineare nella quale la stima $Q(S_t, A_t, \boldsymbol{w}_t)$ è lineare rispetto a $\boldsymbol{w}$. Per ogni coppia-stato azione c'è un vettore reale $\boldsymbol{x}(s,a) = \big(x_1(s,a), x_2(s,a), \dots, x_d(s,a)\big)^T$ con lo stesso numero di componenti di $\boldsymbol{w}$. I metodi lineari approssimano la funzione valore come segue:

$$Q(s,a,\boldsymbol{w}) = \boldsymbol{w}^{\mathbf{T}}\boldsymbol{x}(s,a) = \sum_{i=1}^{d} w_i x_i(s,a). \tag{3.23}$$

Il vettore $\boldsymbol{x}(s,a)$ è chiamato vettore di funzionalità[30] che rappresenta la coppia stato-azione (s,a) ed ogni sua componente $x_i(s,a)\ \epsilon\ \mathbb{R}$. Applicando la funzione di approssimazione lineare al metodo di discesa stocastica del gradiente con rispetto a $\boldsymbol{w}$ si ottiene

[30] Come si vedrà nel prossimo capitolo, nell'applicazione il vettore di funzionalità, di solito viene trasformato mediante l'applicazione di una funzione di *squashing.*

$$\nabla_{\boldsymbol{w}} Q(S_t, A_t, \boldsymbol{w}_t) = \boldsymbol{x}(s, a) \tag{3.24}$$

e quindi l'Equazione 3.20 si trasforma nella seguente forma:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha[U_t - Q(S_t, A_t, \boldsymbol{w}_t)]x(S_t, A_t). \tag{3.25}$$

Più in dettaglio, combinando questi risultati con gli schemi di aggiornamento degli algoritmi appena visti, si ottiene la regola di aggiornamento per SARSA come

$$w_{k+1} \leftarrow w_k + \alpha\big[R_{t+1} + \gamma \boldsymbol{w}_{\boldsymbol{k}}^T \boldsymbol{x}(S_{t+1}, A_{t+1}) - \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{x}(S_t, A_t)\big]\boldsymbol{x}(S_t, A_t) \tag{3.26}$$

e analogamente l'aggiornamento per l'algoritmo Q-*learning* come

$$w_{k+1} \leftarrow w_k + \alpha \left[R_{t+1} + \gamma \max_{a \epsilon A(S_{t+1})} \boldsymbol{w}_{\boldsymbol{k}}^T \boldsymbol{x}(S_{t+1}, a) - \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{x}(S_t, A_t)\right] \boldsymbol{x}(S_t, A_t). \tag{3.27}$$

È possibile provare che, sotto alcune condizioni, l'equazione (3.18) converge ad un vettore $\tilde{\boldsymbol{w}}$ tale per cui $\overline{VE}(\tilde{\boldsymbol{w}})$ ammette il minor errore possibile, portando quindi ad una buona approssimazione. Purtroppo, non è possibile lo stesso per i metodi *off-policy* con approssimazione lineare, dunque non vale per l'equazione (3.26). Per questo motivo è stato sviluppato recentemente un algoritmo, chiamato *Greedy*-GQ, che rappresenta la generalizzazione dell'algoritmo Q-*Learning* e permette di garantire la convergenza attraverso l'approssimazione lineare (Corazza, 2019). Questo nuovo algoritmo utilizza una sequenza di nuovi parametri $\boldsymbol{\theta}_t \epsilon \mathbb{R}$ e un altro parametro *step-size* da aggiungere, β_t, tali che le regole di aggiornamento per i parametri diventano le seguenti:

$$\delta_{t+1} \leftarrow [R_{t+1} + \gamma \boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{x}(S_{t+1}, a') - \boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{x}(S_t, A_t)] \tag{3.28}$$

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha_t[\delta_{t+1}\boldsymbol{x}(S_t, A_t) - \gamma\boldsymbol{\theta}_t^T\boldsymbol{x}(S_t, A_t)]\boldsymbol{x}(S_{t+1}, a') \tag{3.29}$$

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \beta[\delta_{t+1} - \boldsymbol{\theta}_t^T\boldsymbol{x}(S_t, A_t)]\boldsymbol{x}(S_t, A_t) \tag{3.30}$$

con $a' \epsilon \arg \max_{a \epsilon A(S_{t+1})} \boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{x}\,(S_{t+1}, a)$. Si noti che fissando $\boldsymbol{\theta}_0 = 0$ e $\beta_t = 0$, gli algoritmi $(3.28) - (3.30)$ si riducono ad un Q-*Learning*.

Capitolo 4

Applicazione del Reinforcement Learning nei sistemi di trading: gli algoritmi Q-*Learning* e SARSA

Gli argomenti fin qui esposti hanno definito il Reinforcement Learning nella sua struttura teorica, mentre in questo capitolo si propone un'applicazione mediante Matlab a sistemi di trading finanziario. In particolare, l'applicazione considera le serie storiche dei prezzi di cinque titoli finanziari che vengono investigate per mezzo degli algoritmi di Q-*Learning* e SARSA, con l'obiettivo di confrontare le performance di tre funzioni di *reward* per questi due algoritmi. Questi due approcci vengono eseguiti confrontando tre diverse funzioni di *reward*: lo Sharpe ratio, il Burke ratio modificato e il Sortino ratio. In questa applicazione, si utilizza lo Sharpe ratio come una sorta di benchmark per la valutazione delle performance rispetto agli altri due. Questo è possibile grazie al fatto che lo Sharpe *ratio* ha una notevole diffusione in letteratura e rappresenta l'indicatore più utilizzato nel RL.

Il capitolo presenta prima i titoli finanziari con le loro caratteristiche, poi gli elementi che specificano il modello, cioè i tre indicatori utilizzati come funzioni di *reward*, la struttura degli stati, le azioni, e la funzione di *squashing*. In seguito, si anticipano le strutture degli output che vengono invece analizzate e commentate nel prossimo capitolo.

Tra i parametri settati per l'applicazione del modello vi è $k = 500$ che indica il numero di iterazioni che vengono compiute dall'algoritmo. Si ritiene che il numero sia sufficientemente elevato per attribuire significatività statistica allo studio, e risulta un numero che consente agli algoritmi di elaborare i dati senza richiedere tempi di calcolo eccessivi. Il valore di α è pari a 0,05 mentre il fattore

di sconto γ è pari a 0,95. Questi due valori sono molto ricorrenti nella letteratura. In particolare il valore attribuito ad α è sufficientemente piccolo da permettere la convergenza della funzione valore stato-azione al suo valore vero, ma non troppo da renderla troppo lenta. Invece, il fattore di sconto γ assume un valore ragionevole per permettere di dare importanza ai rendimenti futuri, così da ottenere due algoritmi lungimiranti. I parametri che invece vengono studiati in valori e combinazioni diverse sono ε che rappresenta la frequenza di *exploration* compiuta dall'algoritmo, N il numero di rendimenti passati utilizzati per rappresentare gli stati s_t e infine L che rappresenta il numero di rendimenti passati utilizzati nel calcolo delle funzioni di *reward*. Di questi si discutono le scelte nei prossimi paragrafi.

4.1 I titoli e costi di transazione

Il modello utilizza i prezzi di cinque titoli finanziari appartenenti all'indice FTSE.MIB: Amplifon S.p.A., Azimut Holding S.p.A., Banco BPM S.p.A., Campari S.p.A. ed Hera S.p.A. Queste serie di prezzi sono state scaricate da Investing.com selezionando il periodo dal 08/07/2004 al 03/06/2020, per un totale di 4038 prezzi per titolo, pari a quasi 16 anni di trading. Questi titoli sono stati selezionati considerando settori il più possibile diversi tra loro. Amplifon S.p.A. infatti, famosa per gli apparecchi acustici, appartiene al settore Salute. Azimut Holding S.p.A. opera nel settore finanziario, mentre il gruppo bancario Banco BPM S.p.A. appartiene al settore bancario. Campari S.p.A. opera nel settore alimentare e infine Hera S.p.A. fornisce servizi energetici, idrici e ambientali. La tabella 4.1 riassume le statistiche descrittive dei titoli. Dai dati si può osservare che i rendimenti in tutti i titoli presentano una distribuzione leptocurtica[31] e asimmetrica, come è tipico delle serie storiche finanziarie.

[31] Una distribuzione leptocurtica presenta una curva più appuntita rispetto alla curva di una distribuzione normale, i valori centrali sono più frequenti. Nel caso di specie, significa che si presentano con maggiore frequenza (rispetto al caso di una distribuzione normale) rendimenti con valore vicino al valore medio.

	Amplifon	Azimut	Banco BPM	Campari	Hera
Rendimento medio	0,000767400	0,000661621	-0,000509909	0,000629606	0,000311195
Varianza campionaria	0,000499993	0,000583582	0,000978858	0,000277299	0,000270990
Deviazione standard	0,022360534	0,024157441	0,031286709	0,016652296	0,016461780
Massimo	0,218820862	0,153715499	0,347898537	0,106594399	0,152189519
Minimo	-0,194471866	-0,158928571	-0,300791937	-0,161144578	-0,174763033
Curtosi	7,410518361	3,484008265	11,471579943	5,776159211	9,213993725
Asimmetria	0,108301255	0,043541107	0,019651613	0,062662193	-0,284602931
Conteggio	4038	4038	4038	4038	4038

Tabella 4.1: Statistica descrittiva dei titoli finanziari. Analisi dei rendimenti giornalieri percentuali. Periodo: 08/07/2004-03/06/2020. Elaborazione in Excel.

Nell'applicazione degli algoritmi si considerano dei costi di transazione fissi in percentuale, che vengono applicati sia in operazioni di apertura che di chiusura delle posizioni. Non vengono applicati quando l'algoritmo non cambia la posizione, cioè quando due o più azioni consecutive sono uguali. Per entrambi gli algoritmi i costi di transazione sono indicati $\delta = \frac{tc}{2}$ con $tc = 0{,}15\%$. Questi vengono considerati nel calcolo dell´*equity line* netta, cioè

$$equity\ line_{t+1} = equity\ line_t \cdot (1 + gain_{t+1} - \delta|a_t - a_{t-1}|) \qquad (4.1)$$

che rappresenta il valore dell'investimento in $t + 1$ del capitale investito iniziale. Con questa espressione si vede che l'*equity line* lorda dal periodo t viene moltiplicata per il rendimento percentuale realizzato dal sistema di trading al netto del valore dei costi di transazione (se l'espressione $|a_t - a_{t-1}|$ è diversa da zero, cioè se effettivamente sono compiute azioni differenti tra i periodi).

Nelle prossime pagine vengono illustrati i grafici dei prezzi e dei rendimenti logaritmici dei titoli che vengono studiati.

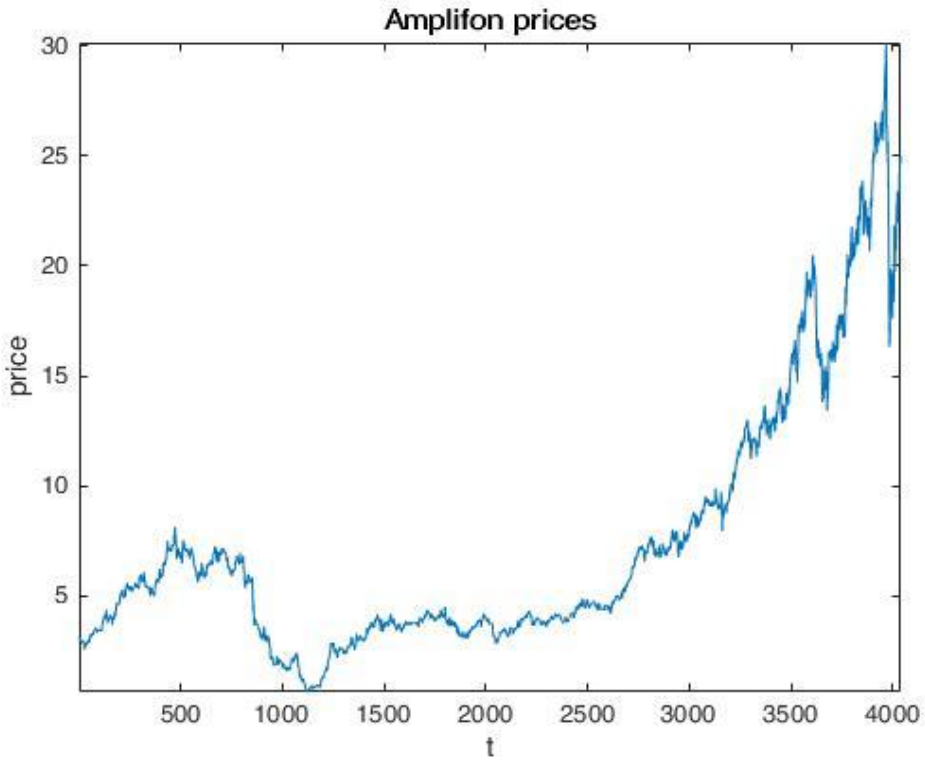

Figura 4.1: Serie dei prezzi Amplifon S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

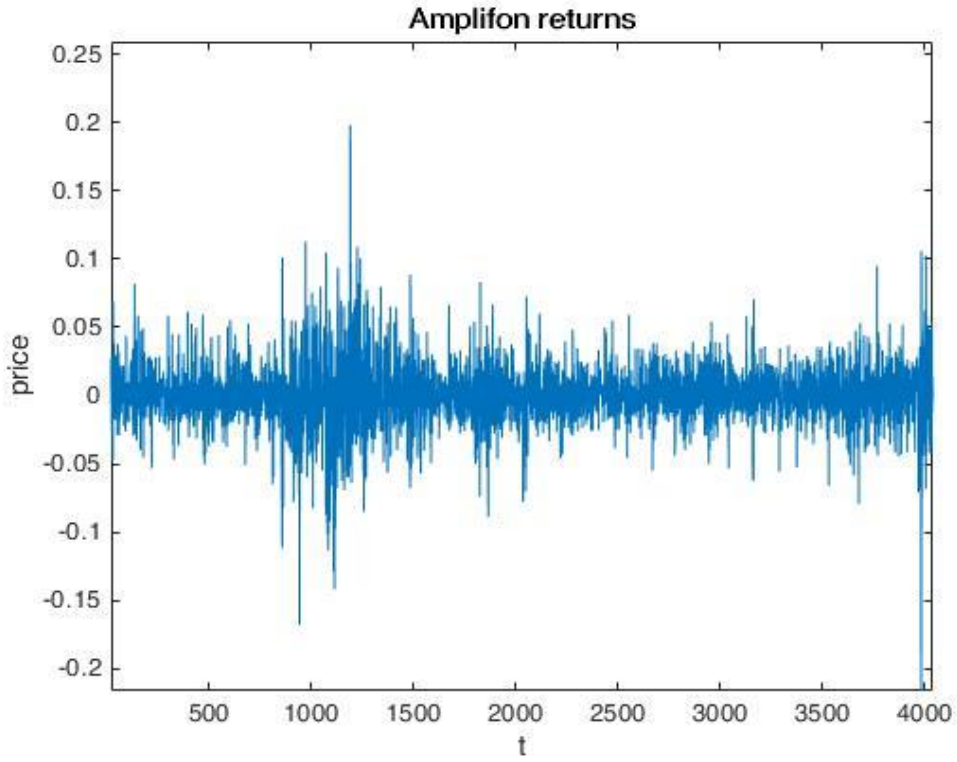

Figura 4.2: Serie dei rendimenti Amplifon S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

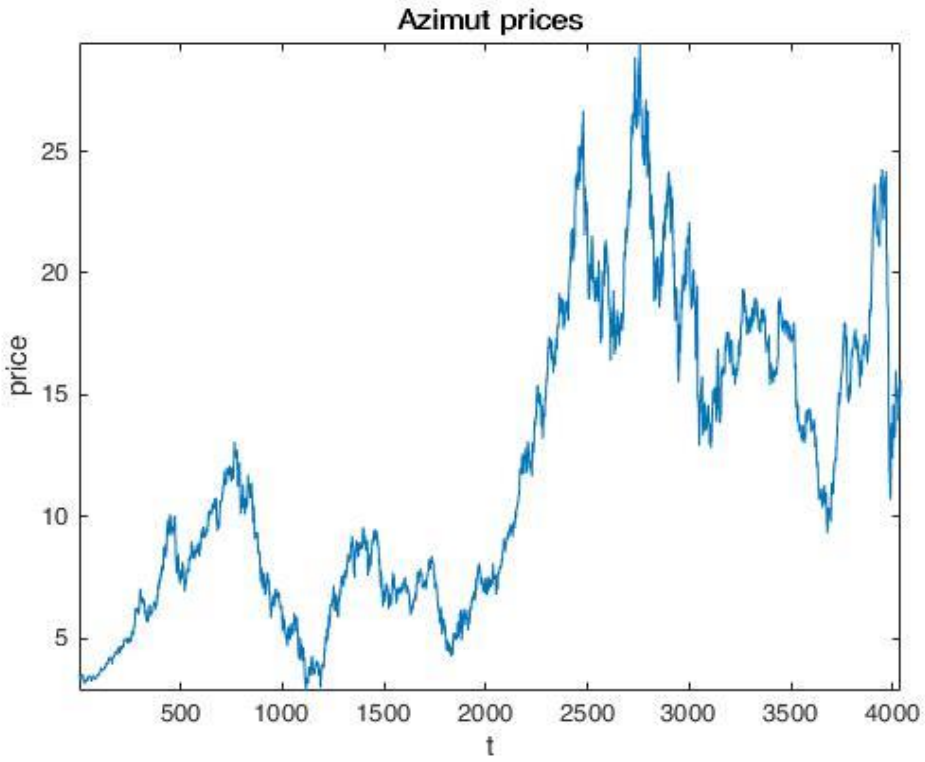

Figura 4.3: Serie dei prezzi Azimut S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

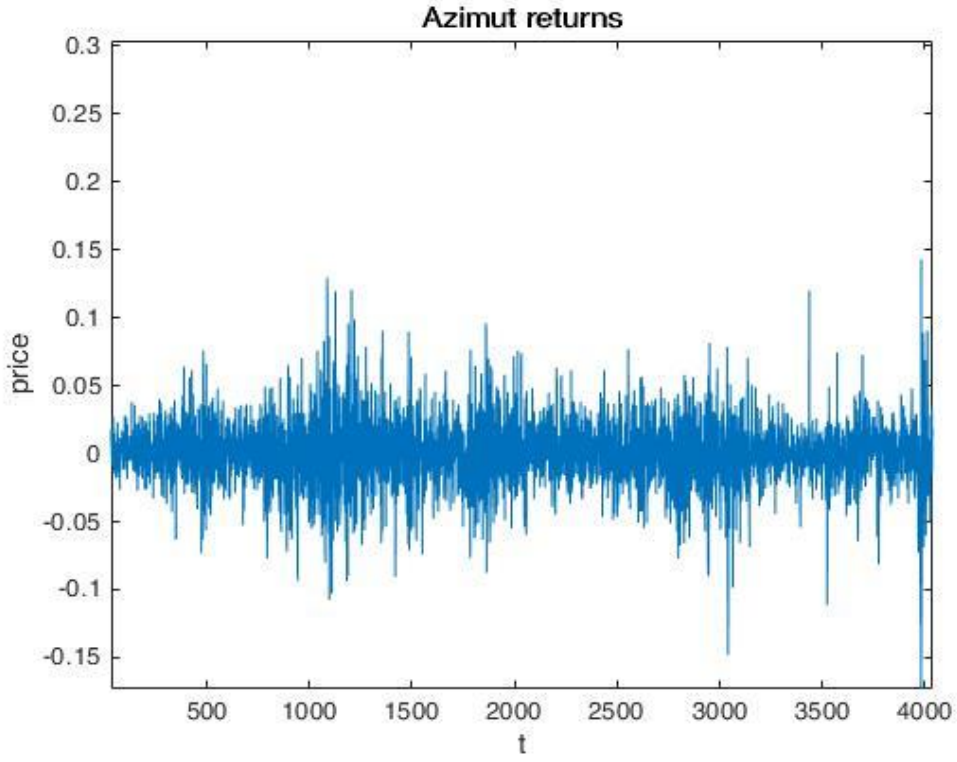

Figura 4.4: Serie dei rendimenti Azimut S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

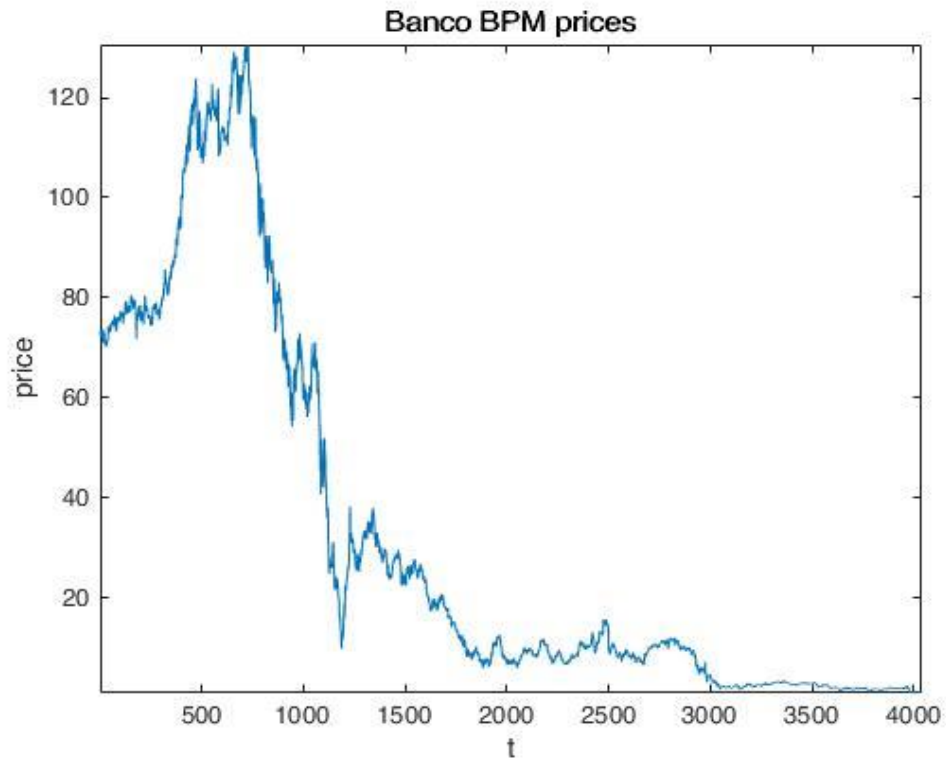

Figura 4.5: Serie dei prezzi Banco BPM S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

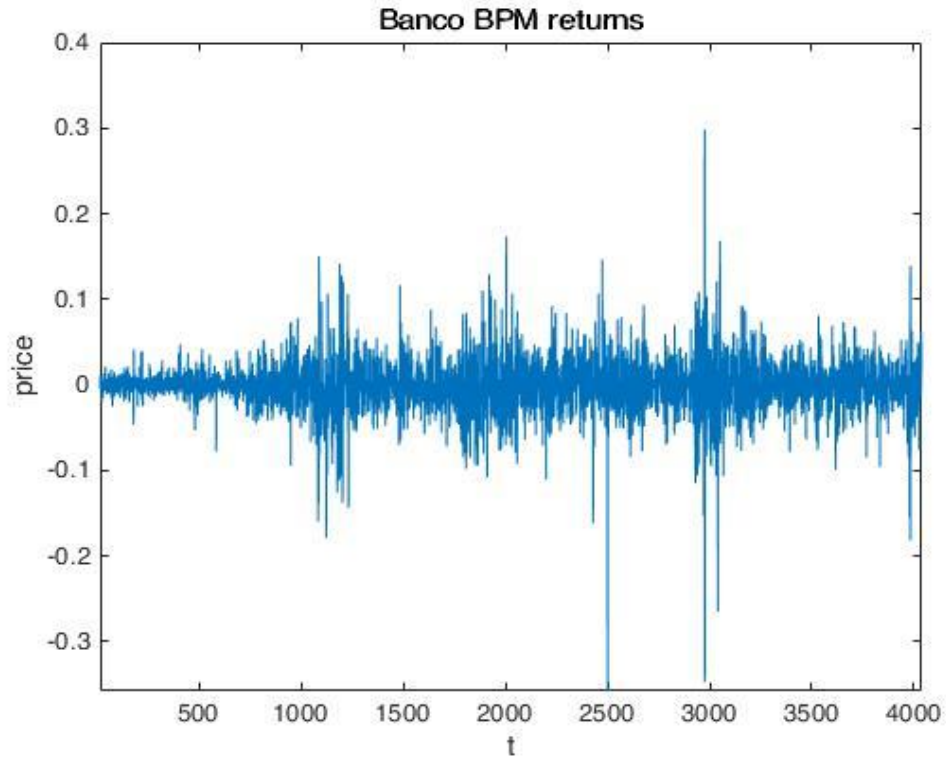

Figura 4.6: Serie dei rendimenti Banco BPM S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

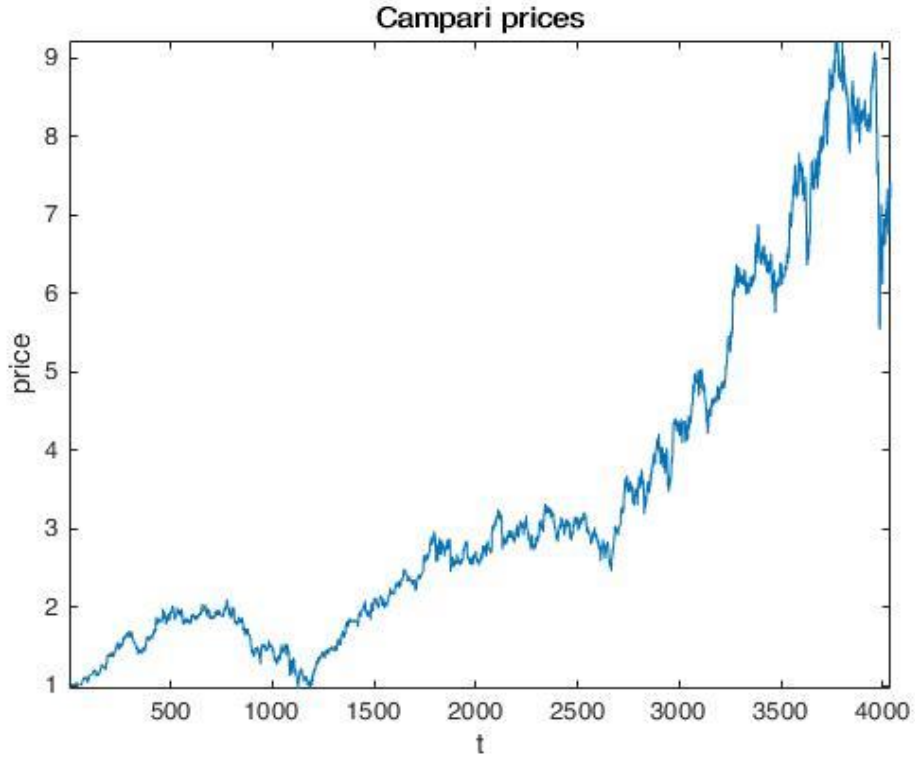

Figura 4.7: : Serie dei prezzi Campari S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

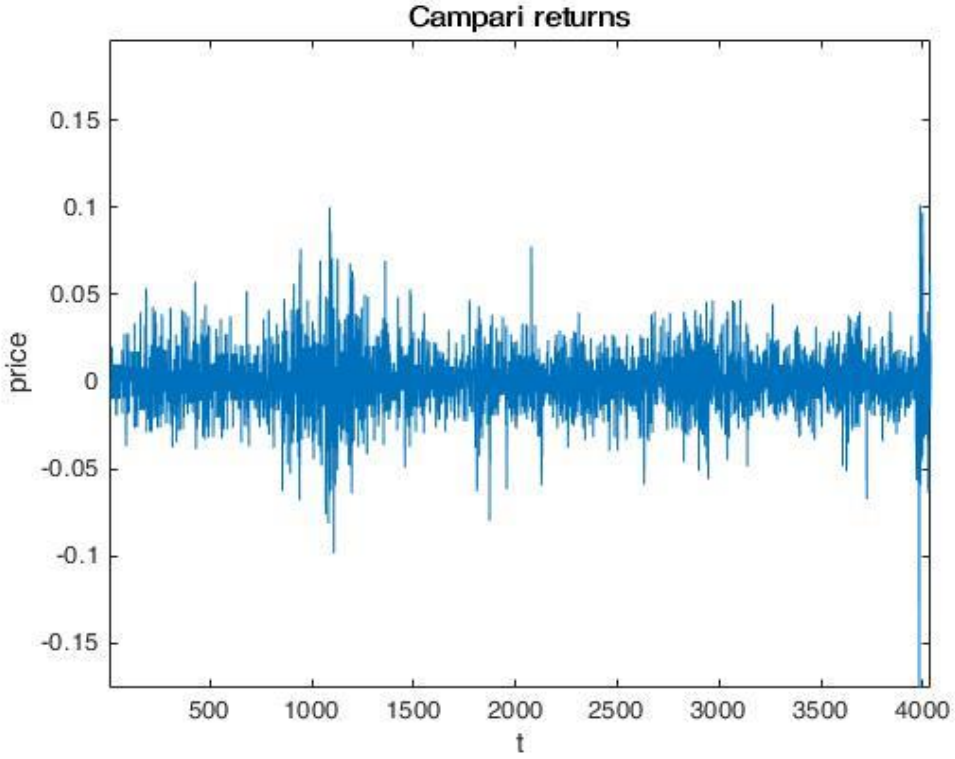

Figura 4.8: : Serie dei rendimenti Campari S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

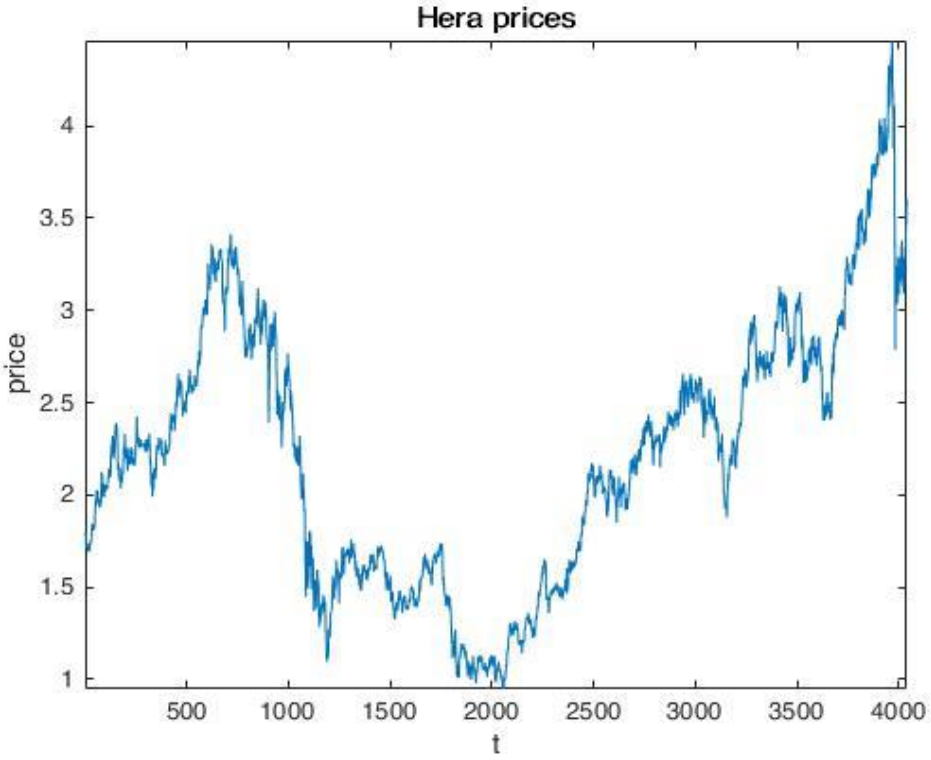

Figura 4.9: Serie dei prezzi HERA S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

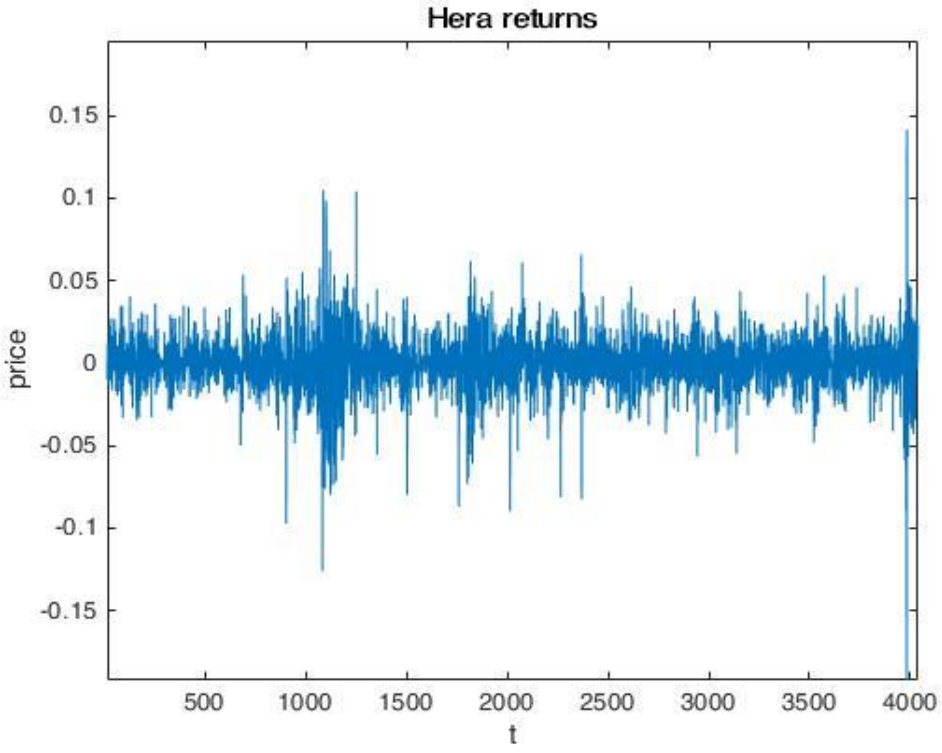

Figura 4.10: Serie dei rendimenti HERA S.p.A. per il periodo 08/07/2004-03/06/2020. Fonte dei dati Investing.com, elaborazione del grafico in Matlab.

4.2 La struttura degli stati s_t

Considerando che si è interessati a verificare le capacità di rendimento dei metodi in esame utilizzando informazioni base per descrivere l'ambiente, la struttura con la quale vengono rappresentati gli stati s_t negli algoritmi consiste semplicemente in un vettore di rendimenti logaritmici al tempo t ed i passati $N-1$ e l'ultima azione intrapresa a_{t-1}. Quindi lo stato s_t è rappresentato dal vettore

$$s_t = (e_{t-N+1}, e_{t-N+2}, \dots, e_t, a_{t-1}) \tag{4.2}$$

con

$$e_t = \log \frac{p_t}{p_{t-1}} \tag{4.3}$$

dove p_t e p_{t-1} sono rispettivamente i prezzi al tempo t e $t-1$. Per entrambi gli algoritmi vengono impostati i valori $N = 1$ e $N = 5$ che rappresentano il numero di giorni di trading dei quali si utilizzano i rendimenti come informazione sull'ambiente. In questo modo si cerca di capire se gli algoritmi sono in grado di reagire più o meno velocemente alle informazioni che ottiene dagli stati, cioè l'ultimo oppure gli ultimi 5 giorni di trading.

4.3. La struttura delle azioni a_t

Nell'applicazione ai sistemi di trading finanziario, le azioni rappresentano le possibili mosse che si possono compiere nei mercati. In questa applicazione le azioni possibili vengono formalizzate nel modo seguente:

$$a_t = \begin{cases} -1 & vendita\ o\ posizione\ "short" \\ 0 & stare\ fuori\ dal\ mercato \\ +1 & acquisto\ o\ posizione\ "long" \end{cases}$$

dove "stare fuori dal mercato" significa chiudere qualsiasi posizione sia aperta sul mercato (se ce ne sono).

Nel RL si è visto in precedenza che risulta necessario assicurare un certo livello di *exploration* delle azioni associate agli stati. Quindi al momento della scelta l'algoritmo seleziona le azioni secondo il criterio *ε-greedy*:

$$\begin{cases} A_t \in arg \max_{a \in A(S_t)} Q(S_t, a) & \text{con probabilità } 1 - \varepsilon_t \\ A(S_t) & \text{con probabilità } \varepsilon_t \end{cases}$$

dove $\varepsilon \in \{5\%, 10\%, 15\%\}$. Il parametro ε definisce la frequenza di esplorazione che compie l'algoritmo. Per esempio, $\varepsilon = 5\%$ significa che con probabilità del 5% l'algoritmo sceglierà un'azione in modo casuale.

4.4 Funzioni di *reward*

Come già anticipato, le funzioni di *reward* utilizzate negli algoritmi sono tre indicatori di misure di performance, cioè lo Sharpe *ratio*, il Burke *ratio* modificato e il Sortino *ratio*. Questi indicatori appartengono alla classe degli indicatori aggiustati per il rischio poiché le loro misure tengono in considerazione sia il rendimento sia il rischio ad esso associato.

Negli algoritmi gli indicatori verranno calcolati sui rendimenti degli ultimi $L = 5$ e $L = 22$ periodi che rappresentano, rispettivamente, gli ultimi 5 e 22 giorni di trading, cioè rendimenti realizzati nell'ultima settimana di borsa e nell'ultimo mese di borsa. La scelta di questi parametri ricade sul fatto che utilizzare i rendimenti degli ultimissimi giorni (meno di una settimana) sarebbe poco significativo per il calcolo degli indicatori (per esempio, si pensi alla significatività di una media campionaria calcolata sugli ultimi due rendimenti), mentre un periodo più lungo di un mese farebbe sì che ad incidere sulle scelte di oggi ci siano rendimenti realizzati più di un mese fa.

4.4.1. Sharpe *ratio*

Lo Sharpe *ratio* è in generale la misura di performance più nota e utilizzata nella gestione di portafogli. Nel RL per rinforzo è l'indice più noto e utilizzato come funzione di *reward*. Per ogni istante t si calcola nel seguente modo:

$$SR_t = \frac{\mathbb{E}_L(g_t) - r_f}{\sigma_L(g_t)} \tag{4.4}$$

dove r_f è il tasso di rendimento *risk-free*[32], $\mathbb{E}_L(g_t)$ è la media campionaria e $\sigma(g_t)$ è la deviazione standard, entrambe calcolate sui rendimenti negli ultimi L giorni di trading. Questo indicatore rapporta il premio per il rischio $\mathbb{E}_L(g_t) - r_f$ al rischio che si sta assumendo per ottenerlo, quindi maggiore è il rendimento per unità di rischio migliore è la performance che si sta realizzando. Se il rapporto è maggiore di 1 significa che l'attività genera rendimenti superiori ai rischi che si sta assumendo e quindi è profittevole. Al contrario quando l'indice è minore di 1 significa che il rendimento non ricompensa a sufficienza il rischio che si assume.

Il limite principale dello Sharpe *ratio* è che il rischio si esprime con la deviazione standard, dunque come grado di dispersione dei rendimenti rispetto al valore medio, ma non è però in grado di distinguere se siano rendimenti superiori o inferiori alla media, che quindi vengono valutati allo stesso modo. Chiaramente nel mondo finanziario uno scostamento positivo dalla media è un rendimento desiderato, non è corretto considerarlo al pari di uno scostamento negativo che invece fa calare le performance. Per ovviare a questo problema sono stati sviluppati altri indicatori, due dei quali sono i seguenti.

4.2.2. Burke *ratio*

Il Burke *ratio* è una misura di performance aggiustata per il rischio sviluppata da Burke nel 1994 nel paper "*A sharper Sharpe ratio*". Con questo indicatore si vuole misurare le performance dei rendimenti in relazione al rischio definito come

[32] Come avviene spesso, anche in questo elaborato si assume $r_f = 0$ poiché r_f rappresenta il tasso di rendimento di un'attività priva di rischio quindi, per definizione, prossimo allo zero.

drawdown, ovvero l'ammontare di perdita in corso rispetto all'ultimo picco.[33] Più precisamente un *drawdown* misura la quantità di denaro persa per fare trading, calcolato come differenza tra un picco di massimo e un picco di minimo del valore del prezzo. Questa misura poi viene utilizzata come denominatore dal ratio rappresentando il rischio associato alla performance realizzata, cioè alla media dei rendimenti. In formula il Burke ratio si esprime come segue:

$$BR_t = \frac{\mathbb{E}_L(g_t) - r_f}{\sqrt{\sum_{i=1}^{d} DD_i^2}} \tag{4.5}$$

dove d indica il numero di *drawdown* e DD_i l'i-esimo *drawdown*, ottenuti negli ultimi L giorni di trading. L'espressione assomiglia allo Sharpe *ratio*, con la differenza che al denominatore vi è una diversa misura di rischio che considera solo gli scostamenti negativi poiché valuta l'ampiezza di perdita in cui un *asset* incorre, calcolata come distanza percentuale dal picco precedente. Ad esempio, si immagini di acquistare un titolo per 100€ al tempo t, che dopo al tempo $t+1$ aumenta di valore arrivando a 110€ (picco), per poi scendere fino a 80€ al tempo $t+2$. Il *drawdown* relativo al periodo $[t; t+2]$ è dunque $\left(\frac{110€-80€}{110€}\right) = 0{,}273$

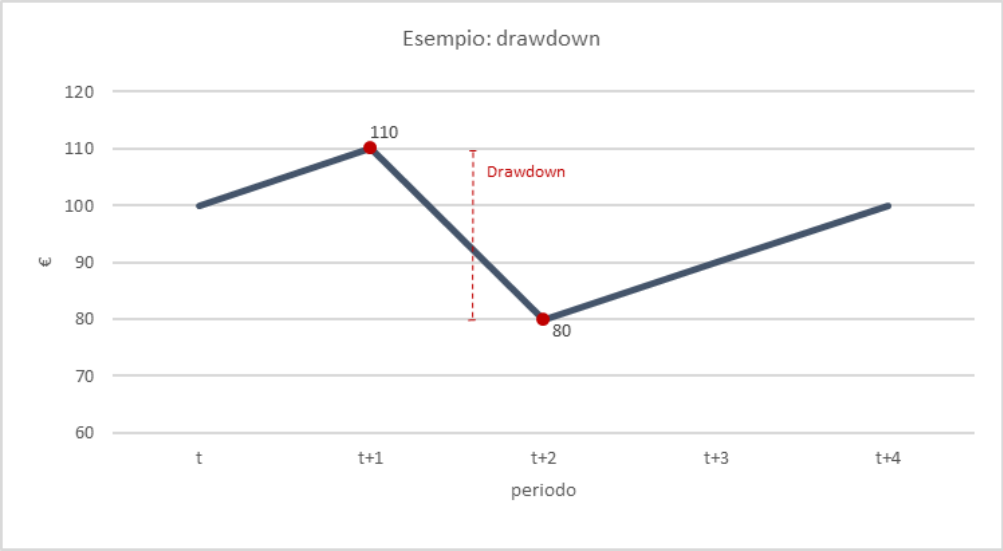

Figura 4.1: Esempio di drawdown. Elaborazione in Excel.

cioè il 27,3%. Si veda la Figura 4.11 come illustrazione esemplificativa.

[33] https://breakingdownfinance.com

Nell'applicazione in questo elaborato si utilizza la versione del Burke *ratio* modificata, ovvero la versione che considera anche il numero di *drawdown* che si realizzano nel periodo in considerazione. La formula diventa quindi:

$$BR_t = \frac{\mathbb{E}_L(g_t) - r_f}{\sqrt{\sum_{i=1}^{L} \frac{DD_i^2}{n}}} \tag{4.6}$$

con n il numero di *drawdown* considerati nel periodo L. In questo modo si ottiene una sorta di *drawdown* medio. Si noti che, a parità di tutte le altre variabili, all'aumentare di n aumenta anche il valore di BR_t. Se per esempio per lo stesso L si realizzano in un caso $n = 1$ e in un altro $n = 10$ con un DD dello stesso valore finale, allora il ratio attribuisce a $n = 10$ un valore maggiore, poiché considera che si sono realizzati tanti "piccoli" *drawdown*, mentre con $n = 1$ si è realizzato solo un DD ma con un valore di perdita maggiore, quindi più rischioso.

4.4.3 Sortino *ratio*

Il Sortino *ratio* è un'altra misura di performance aggiustata per il rischio che considera solo scostamenti negativi dei rendimenti. Come il Burke *ratio*, anche il Sortino è una versione modificata dello Sharpe *ratio*. La formula è la seguente:

$$SOR_t = \frac{\mathbb{E}_L(g_t) - r_f}{\sqrt{\sigma_L^2[\min(0, g_t - \bar{g})]}}. \tag{4.7}$$

dove $\mathbb{E}_L(g_t)$ e $\sigma(r_t)$ sono rispettivamente la media campionaria e la deviazione standard, calcolata come radice della varianza sul minimo tra 0 e lo scostamento dei rendimenti g_t dalla media $\bar{g}$. Entrambi si riferiscono agli ultimi L giorni di trading. La misura al denominatore consiste in un *downside risk*, rappresentata dalla deviazione standard campionaria che considera solo gli scostamenti negativi dalla media del periodo. Anche il Sortino, come il Burke *ratio*, dunque permette

di considerare soltanto gli scostamenti negativi dei rendimenti, rendendosi più appropriato a sistemi di trading.

4.4 Funzione di *squashing*

La funzione di *squashing* viene utilizzata per trasformare gli input che definiscono gli stati s_t in modo da rendere più sensibile il processo di calcolo della funzione di valore $Q(s_t, a_t)$ (Corazza et al., 2015), per mezzo della forma che assume questa funzione. Infatti, la funzione che viene considerata nell'applicazione è la seguente funzione logistica:

$$\phi(x) = \frac{a}{1 + be^{-cx}} - d \tag{4.7}$$

con $a = 2, b = 1, c = 10^{15}$ e $d = -1$. Questo tipo di settaggio rende la funzione di *squashing* simile ad una funzione tangente iperbolica, permettendo di amplificare la sensibilità dell'agente. Infatti, valori apparentemente vicini tra loro verranno considerati più distanti dopo la trasformazione facendo sì che anche piccole variazioni degli stati permettano di dare un segnale. Questo aiuta l'agente ad essere più capace di distinguere piccole variazioni di rendimento e di trasformarle in informazioni utili per compiere le scelte.

4.5 Il meccanismo di calcolo degli algoritmi e gli output risultanti

Riassumendo, il settaggio dei parametri per l'applicazione agli algoritmi Q-*Learning* e SARSA è:

- $\alpha = 0{,}05$;
- $\gamma = 0{,}95$;
- $tc = 0{,}15\%$ ovvero $\delta = 0{,}075\%$ per ogni entrata o uscita di una posizione;
- $\varepsilon \in \{5\%, 10\%, 15\%\}$;
- $N = 1$ e $N = 5$;
- $L = 22$ e $L = 5$;
- $z = 500$.

Ogni volta che si effettua un'applicazione, questa richiede l'utilizzo di tutte le possibili combinazioni di questi parametri, per due diversi algoritmi per ognuno dei quali il software compie 500 iterazioni. È chiaro dunque che la mole di calcolo per un singolo titolo è notevole e richiede un certo tempo.

All'inizio dell'applicazione vengono costruiti i vettori di pesi θ usati come parametri e i vettori di stato s_t come descritti sopra. Per ogni iterazione k, viene generato un valore random. Se questo valore è minore di ε si compie un'azione esplorativa per effettuare la quale viene generato un altro numero random tra 1 e 3 che corrispondono alle tre azioni possibili (come descritto sopra) e se ne calcola la funzione valore stato-azione (sia per Q-*Learning* che SARSA) per mezzo della funzione logistica. Se invece il valore random è maggiore di ε si sceglie l'azione che massimizza il valore della funzione valore stato-azione associato alle possibili azioni, cioè

$$[maxQL, rand(3)]) = max\left[Q\left(s_t, a_{t,-1}, \theta_t\right) \;\; Q(s_t, a_{t,0}, \theta_t) \;\; Q(s_t, a_{t,+1}, \theta_t)\right]$$
$$[maxSARSA, rand(3)]) = max\left[Q\left(s_t, a_{t,-1}, \theta_t\right) \;\; Q(s_t, a_{t,0}, \theta_t) \;\; Q(s_t, a_{t,+1}, \theta_t)\right]$$

dove i valori che descrivono le funzioni valore stato-azioni (gli stati s_t di riferimento e il vettore θ_t) sono diversi in base all'algoritmo che si sta studiando.

Selezionata l'azione, si calcola il valore del rendimento realizzato attraverso la formula $g_{t+1} = e_{t+1}a_t$, che moltiplica il logaritmo della differenza dei prezzi al netto dell'eventuale costo di transazione, cioè il rendimento ottenuto, per l'azione che corrisponde alla posizione precedentemente assunta, dunque $a_t = (-1,0,1)$. Con questi rendimenti si calcolano le *equity lines* per algoritmo, per ogni iterazione k e ogni periodo $t+1$, che rappresentano il valore dell'investimento espresso (in questo caso) in euro:

$$equity\ line_{t+1,k}^{QL} = equity\ line_{t,k}^{QL} \cdot (1 + g_{t+1}^{QL})$$
$$equity\ line_{t+1,k}^{SARSA} = equity\ line_{t,k}^{SARSA} \cdot (1 + g_{t+1}^{SARSA}).$$

Poi, al termine delle iterazioni, si calcola la funzione di *reward* sugli ultimi *L* rendimenti realizzati.

Una volta giunti a questo punto, si calcolano gli elementi necessari per l'aggiornamento del vettore θ_t sia per Q-*Learning* sia per SARSA, cioè il valore dello stato successivo s_{t+1}, il TD error $r_{t+1} + \gamma Q_k(s_{t+1}, a_t) - Q_k(s_t, a_t)$ e il gradiente $\nabla_{\theta_t} Q$. A questo punto si compie l'aggiornamento del vettore dei pesi $\theta_{t+1} = \theta_t + \alpha \delta_t \nabla_{\theta_t} Q$. Per evitare confusione, si noti che con g_{t+1} si intende il rendimento realizzato calcolato come logaritmo della differenza dei prezzi al netto degli eventuali costi di transazione, mentre il *reward* r_{t+1} è la misura di performance che si sta utilizzando, per calcolare la quale si utilizzano gli ultimi rendimenti (anche g_{t+1}).

Una volta ottenute tutte le 500 iterazioni per ogni periodo t, si devono tradurre tutte le azioni realizzate dalle iterazioni in un'unica azione da associare ad ogni istante t. Per questo motivo si compie la media di tutte le azioni che gli algoritmi hanno selezionato, cioè $\bar{a}_t = \frac{\sum_{i=1}^{k} a_{t,i}}{k}$ sia per Q-*Learning* che per SARSA. Il valore che risulta, viene tradotto in un segnale operativo attraverso la seguente logica:

$$a_t = \begin{cases} -1 < \bar{a}_t < -\frac{1}{3} \; allora \; a_t = -1 \; posizione \; di \; vendita \; o \; "short" \\ -\frac{1}{3} \leq \bar{a}_t \leq \frac{1}{3} \quad allora \; a_t = 0 \; stare \; fuori \; dal \; mercato \\ \frac{1}{3} < \bar{a}_t < +1 \; allora \; a_t = +1 \; posizione \; di \; acquisto \; o \; "long" \end{cases}$$

Una volta definite le azioni per ogni istante t, si calcolano tutte le statistiche e le informazioni differenti per Q-*Learning* e SARSA: i rendimenti e le *equity lines* e tutte le statistiche finali come l'ammontare di capitale finale, i rendimenti realizzati e il numero di operazioni compiute all'anno dall'algoritmo. Tutte queste voci, a parte il numero di operazioni, vengono calcolate al lordo e al netto dei costi di transazione.

Inoltre, vengono create delle tabelle con le seguenti informazioni: capitale finale lordo e netto (€), il rendimento giornaliero medio annualizzato lordo e netto (%), il rendimento annuale medio lordo e netto (%), la percentuale lorda e netta in

cui il capitale si trova sopra o in corrispondenza della soglia dei 100€[34] e il numero di operazioni annue compiute dall'algoritmo. Tutte queste informazioni sono ottenuto per singolo algoritmo, singolo titolo e per ognuna delle possibili combinazioni di L, N ed ε.

Si ottengono anche alcuni grafici. Il primo tipo di grafico raccoglie le *equity lines* di tutte le 500 iterazioni, separate per algoritmo. Con questo è possibile vedere se le iterazioni hanno un grado di dispersione più o meno ampio e più o meno positivo, oppure se sono in generale traiettorie simili oppure molto diverse. Dalla media di queste *equity lines*, si ottiene anche un grafico per algoritmo con la traiettoria dell'*equity line* media. Di seguito un esempio con le Figure 4.12 e 4.13. Il secondo tipo di grafico, sempre diviso per Q-*Learning* e per SARSA, raccoglie al suo interno tre pannelli, che illustrano: l'andamento del prezzo del titolo, le azioni che il software ha compiuto e le *equity lines* lorde e nette che derivano dalle operazioni. Un esempio di questo grafico in Figura 4.14.

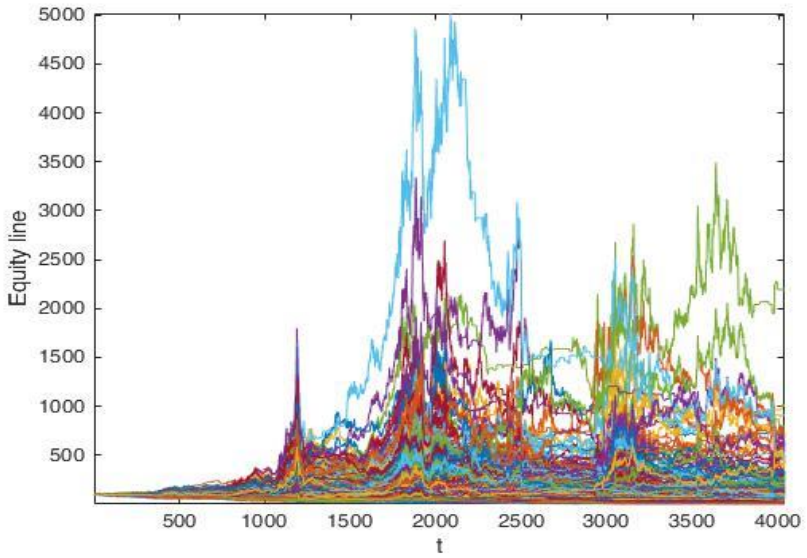

Figura 4.12: Amplifon S.p.A., 500 equity lines per Q-Learning Sharpe con $L = 5$, $N = 1$, $\varepsilon = 10$.

[34] Questa informazione è molto interessante perché, a differenza del capitale finale realizzato che può essere più o meno positivo sulla base di quando viene interrotto l'esperimento, indica la probabilità con cui è possibile ottenere un guadagno qualora l'investimento venga interrotto in un momento qualsiasi del periodo. Per esempio, se il valore di "Perc. over =100" é 95,02 significa che se si interrompe l´investimento nel 92,05% dei casi si avrà un capitale finale almeno pari a 100€, quindi uguale o superiore.

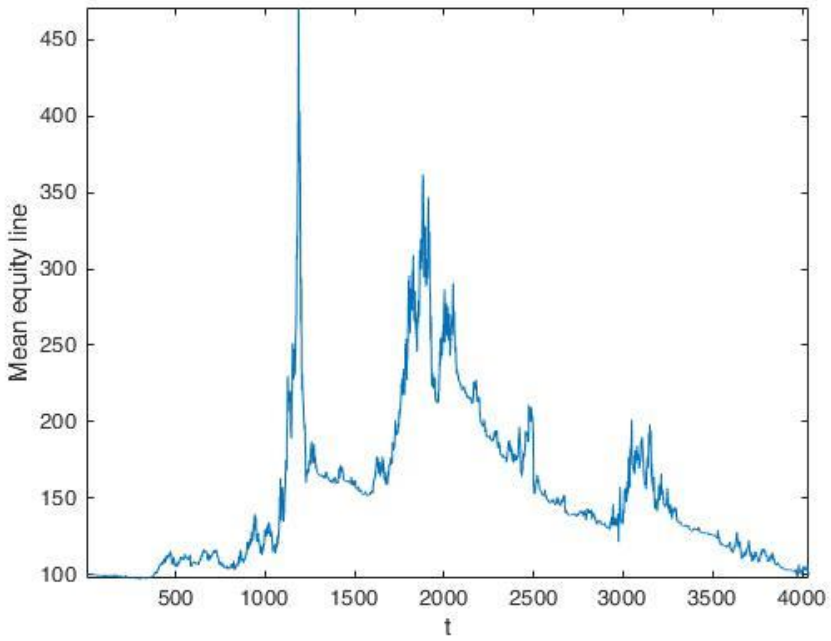

Figura 4.13: Amplifon S.p.A., media delle equity lines in Figura 4.12 per Q-Learning Sharpe con $L = 5$, $N = 1$, $\varepsilon = 10$.

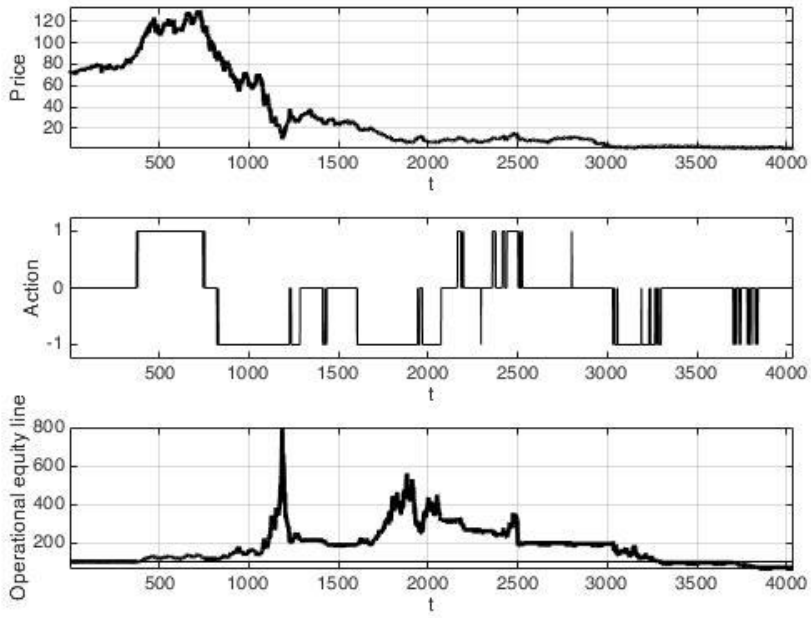

Figura 4.14: Amplifon S.p.A, Q-Learning, Sharpe con L=5, N=1, ε=10. Dall'alto: prezzi del titolo Amplifon S.p.A., azioni intraprese dal software ed equity line realizzata.

Capitolo 5

Applicazione del Reinforcement Learning ai sistemi di trading finanziari: osservazioni e risultati

5.1 Risultati

In questo capitolo si presentano e discutono i risultati ottenuti dall'applicazione degli algoritmi di Reinforcement Learning presentati nel Capitolo 4. Gli obiettivi dell'applicazione che si cerca di raggiungere con questa analisi dei dati sono:

- Confrontare le performance tra i due algoritmi, Q-Learning e SARSA, come sistemi automatici di trading;
- Confrontare le misure di performance tra Sharpe ratio, diffuso in letteratura, con le alternative proposte dall'elaborato: Sortino ratio e Burke ratio;
- Valutare il comportamento degli algoritmi e/o dei ratio al variare del settaggio dei parametri N, L, ε;
- Individuare, se possibile, il miglior metodo tra le possibili combinazioni di algoritmi, funzioni di reward e parametri in termini di profitto e di efficienza nella costruzione di una strategia di trading.

Alla fine del capitolo, dopo un ulteriore approfondimento realizzato sulla base delle prime osservazioni, l'ultimo paragrafo riassume i risultati della discussione e risponde agli obiettivi appena puntualizzati.

5.1.1 Amplifon S.p.A.

Nella Tabella 5.1 si riportano i dati relativi ad Amplifon S.p.A. ottenuti con QL per $N = 1$. Si può notare che il capitale finale ottenuto con Sharpe ratio è maggiore di 100 per tutte le combinazioni, per di più con valori di performance molto alti, tanto che il rendimento annuo netto medio è pari a 5,052%. Il maggior valore lo raggiunge con $L = 22$ e $\varepsilon = 10\%$ con un capitale finale netto di 390,063€ a cui si associa un rendimento netto annuo di 8,868% che risulta essere il rendimento maggiore in assoluto tra tutti i titoli e tutti gli algoritmi. Con il Burke ratio ottiene un rendimento negativo per $L = 5$ $\varepsilon = 5$ pari a $-4{,}398\%$ che corrisponde ad un capitale finale di 56,487€. Gli altri rendimenti ottenuti con Burke, invece, raggiungono risultati in linea con quelli di Sharpe, in particolare con $L = 22$ e $\varepsilon = 10\%$ dove ottiene come per Sharpe il rendimento più alto, pari a 7,201% che corrisponde ad un capitale finale netto di 304,648€. Nel caso di Sortino ratio per $L = 5$ si comporta come Sharpe e Burke, quindi rendimenti positivi soprattutto per $\varepsilon = 10\%$. Per $L = 22$ invece, $\varepsilon = 5\%$ risulta un ottimo

N=1	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	L	ε (%)	*G €*	***N €***	*G %*	***N %***	*G %*	***N %***	*G %*	***N %***	*#*
SR	5	5	153,734	**141,881**	0,011	**0,009**	2,721	**2,208**	88,308	**85,682**	**6,7**
		10	257,496	**226,818**	0,023	**0,020**	6,082	**5,245**	91,305	**90,587**	**10,5**
		15	263,854	**233,857**	0,024	**0,021**	6,244	**5,446**	84,642	**83,527**	**10,1**
	22	5	199,210	**186,899**	0,017	**0,015**	4,396	**3,981**	75,675	**74,808**	**5,3**
		10	421,995	**390,063**	0,036	**0,034**	9,404	**8,868**	99,529	**99,529**	**6,6**
		15	246,718	**204,353**	0,022	**0,018**	5,799	**4,562**	99,950	**99,554**	**15,7**
BR	5	5	56,487	**48,648**	-0,014	**-0,018**	-3,503	**-4,398**	0,694	**0,694**	**12,4**
		10	235,081	**207,715**	0,021	**0,018**	5,481	**4,669**	78,524	**78,003**	**10,3**
		15	188,573	**166,120**	0,016	**0,013**	4,039	**3,219**	78,722	**78,003**	**10,5**
	22	5	144,561	**127,329**	0,009	**0,006**	2,327	**1,520**	25,811	**24,251**	**10,5**
		10	334,669	**304,648**	0,030	**0,028**	7,832	**7,201**	95,046	**94,773**	**7,8**
		15	218,387	**187,649**	0,019	**0,016**	4,997	**4,007**	92,247	**91,925**	**12,6**
SOR	5	5	219,027	**193,530**	0,019	**0,016**	5,016	**4,208**	77,558	**52,192**	**10,3**
		10	241,093	**208,289**	0,022	**0,018**	5,647	**4,687**	77,310	**48,353**	**12,2**
		15	222,648	**201,207**	0,020	**0,017**	5,123	**4,461**	77,359	**77,260**	**8,4**
	22	5	312,843	**282,770**	0,028	**0,026**	7,379	**6,704**	99,926	**99,926**	**8,4**
		10	66,388	**55,601**	-0,010	**-0,015**	-2,525	**-3,598**	32,698	**15,209**	**14,7**
		15	65,589	**56,533**	-0,010	**-0,014**	-2,598	**-3,498**	24,820	**21,848**	**12,4**

Tabella 5.1: Amplifon S.p.A. per QL con N=1

rendimento annuo netto pari a $6{,}704\%$, mentre sono negativi i rendimenti per $\varepsilon \in \{10\%, 15\%\}$ che sono, rispettivamente, pari a $-3{,}598\%$ e $-3{,}498\%$ cioè 66,388€ e 65,589€. Il numero di operazioni è in generale basso con tutte le funzioni di reward, con il valore maggiore ottenuto per Sortino $L = 22$ e $\varepsilon = 10\%$. Le percentuali di volte in cui l'*equity line* è superiore o uguale a 100 (colonna "perc over 100") è in generale alta per tutte le combinazioni di Sharpe. Per Burke ratio è alta con $\varepsilon \in \{10\%, 15\%\}$, ma con $\varepsilon = 5\%$ (per entrambi i valori di L) si abbassa molto. In particolare $L = 5$ e $\varepsilon = 5\%$ ha una percentuale over 100 pari a $0{,}694\%$, che significa che nel $99{,}306\%$ del tempo di investimento l'*equity line* è in perdita. Con Sortino si hanno percentuali over 100 basse quando la performance è negativa, dunque per $L = 22$ ε e$\in \{10\%, 15\%\}$.

La seconda tabella raccoglie gli output relativi ad Amplifon S.p.A. ottenuti con l'algoritmo QL con $N = 5$. Nel caso di Sharpe i rendimenti sono positivi e con capitale finale netto, come nella tabella precedente per $N = 1$. Le percentuali over 100 sono anche in questo caso alte e positive, con valori un po' più bassi nel caso di $\varepsilon = 5\%$, mentre il numero di operazioni annue è leggermente più

N=5	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	L	ε (%)	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	#
SR	5	5	197,657	**182,680**	0,017	**0,015**	4,349	**3,837**	78,726	**76,345**	**6,6**
		10	234,468	**209,632**	0,021	**0,018**	5,469	**4,734**	95,016	**91,495**	**9,3**
		15	312,671	**279,071**	0,028	**0,025**	7,383	**6,623**	83,982	**82,643**	**9,4**
	22	5	327,253	**300,636**	0,029	**0,027**	7,689	**7,120**	83,387	**78,800**	**7,1**
		10	298,360	**268,811**	0,027	**0,025**	7,069	**6,374**	99,504	**99,331**	**8,7**
		15	297,813	**254,210**	0,027	**0,023**	7,057	**6,003**	98,140	**97,396**	**13,2**
BR	5	5	113,995	**104,116**	0,003	**0,001**	0,822	**0,252**	8,802	**5,802**	**7,6**
		10	105,278	**94,140**	0,001	**-0,001**	0,322	**-0,377**	13,637	**9,745**	**9,3**
		15	202,733	**180,751**	0,018	**0,015**	4,515	**3,768**	80,238	**79,916**	**9,6**
	22	5	175,226	**164,884**	0,014	**0,012**	3,567	**3,174**	31,341	**29,928**	**5,1**
		10	202,388	**176,975**	0,017	**0,014**	4,504	**3,631**	89,512	**81,255**	**11,2**
		15	351,665	**304,933**	0,031	**0,028**	8,174	**7,215**	95,983	**95,834**	**11,9**
SOR	5	5	231,058	**215,511**	0,021	**0,019**	5,372	**4,915**	47,830	**44,657**	**5,8**
		10	202,537	**181,078**	0,018	**0,015**	4,509	**3,780**	81,453	**81,255**	**9,3**
		15	282,591	**251,613**	0,026	**0,023**	6,706	**5,935**	89,115	**88,916**	**9,7**
	22	5	144,883	**131,239**	0,009	**0,007**	2,344	**1,713**	99,901	**99,876**	**8,2**
		10	82,352	**69,835**	-0,005	**-0,009**	-1,206	**-2,218**	88,321	**81,379**	**13,7**
		15	128,184	**116,999**	0,006	**0,004**	1,564	**0,986**	99,975	**99,802**	**7,6**

Tabella 5.2: Amplifon S.p.A. per QL con N=5

basso. Con Burke i rendimenti sono negativi solo con $L = 5$ e $\varepsilon = 10\%$ dove l'algoritmo termina con un capitale di $94{,}140€$ e un rendimento annuo netto di $-0{,}377\%$ a cui si associa una percentuale over 100 molto bassa, di circa il 10%. Ancora più bassa la percentuale nel caso $\varepsilon = 5\%$ che è pari a $5{,}802\%$ anche se termina con un capitale finale di $104{,}116€$. Negli altri casi i rendimenti sono simili al caso $N = 1$, quindi positivi ma minori di Sharpe ad eccezione di $L = 22$ e $\varepsilon = 10\%$ con cui Burke ottiene il valore più alto della tabella con rendimento annuo netto di $7{,}215\%$. Il risultato è comunque molto vicino al rendimento di $7{,}120\%$ ottenuto da Sharpe con $L = 22$ e $\varepsilon = 15\%$, secondo per performance. Sortino risulta in linea con quanto accade con $N = 1$: i risultati di $L = 22$ e $\varepsilon \in \{10\%, 15\%\}$ sono minori, con la differenza che con $\varepsilon = 15\%$ questa volta il rendimento è positivo e la percentuale over 100 è prossima al 100.

Dal confronto tra i risultati per $N = 1$ e $N = 5$ si evince che con QL i risultati sono consistenti: la tendenza generale è quella di rendimenti positivi, alte percentuali over 100, in particolare per Sharpe che ottiene il rendimento più alto dell'intera applicazione. Il risultato negativo di Burke può essere dovuto all'andamento dei prezzi del titolo Amplifon S.p.A. che, soprattutto nel primo periodo, è privo di trend ma oscilla senza una precisa tendenza rialzista o ribassista. Questo comporta che i segnali operativi non siano interpretati in modo efficiente e non si ottenga un apprendimento crescente dell'algoritmo nel periodo di investimento. Per campire meglio si veda la seguente Figura 5.1. Il pannello più in alto nei grafici illustra l'andamento del prezzo, quello in mezzo le azioni intraprese mentre l'ultimo descrive l'*equity line* risultante (lorda quando la linea è continua, netta quando è tratteggiata). Si nota che nel primo periodo fino a $t = 1200$ circa, i due ratio (Burke in alto, Sharpe in basso) reagiscono in modo diverso all'oscillazione del prezzo che non assume posizioni chiare di trend in rialzo o ribasso. In questo periodo Burke performa male accumulando perdite e senza riuscire ad invertire l'andamento, mentre Sharpe sebbene timidamente, riesce a rimanere in un intorno del valore di investimento iniziale, cioè $100€$. Questa differenza potrebbe essere dovuta alla diversa natura della volatilità nei due ratio: in Sharpe infatti si considerano scostamenti positivi quanto negativi rispetto alla media, mentre in Burke si considerano solo quelli negativi. Questo

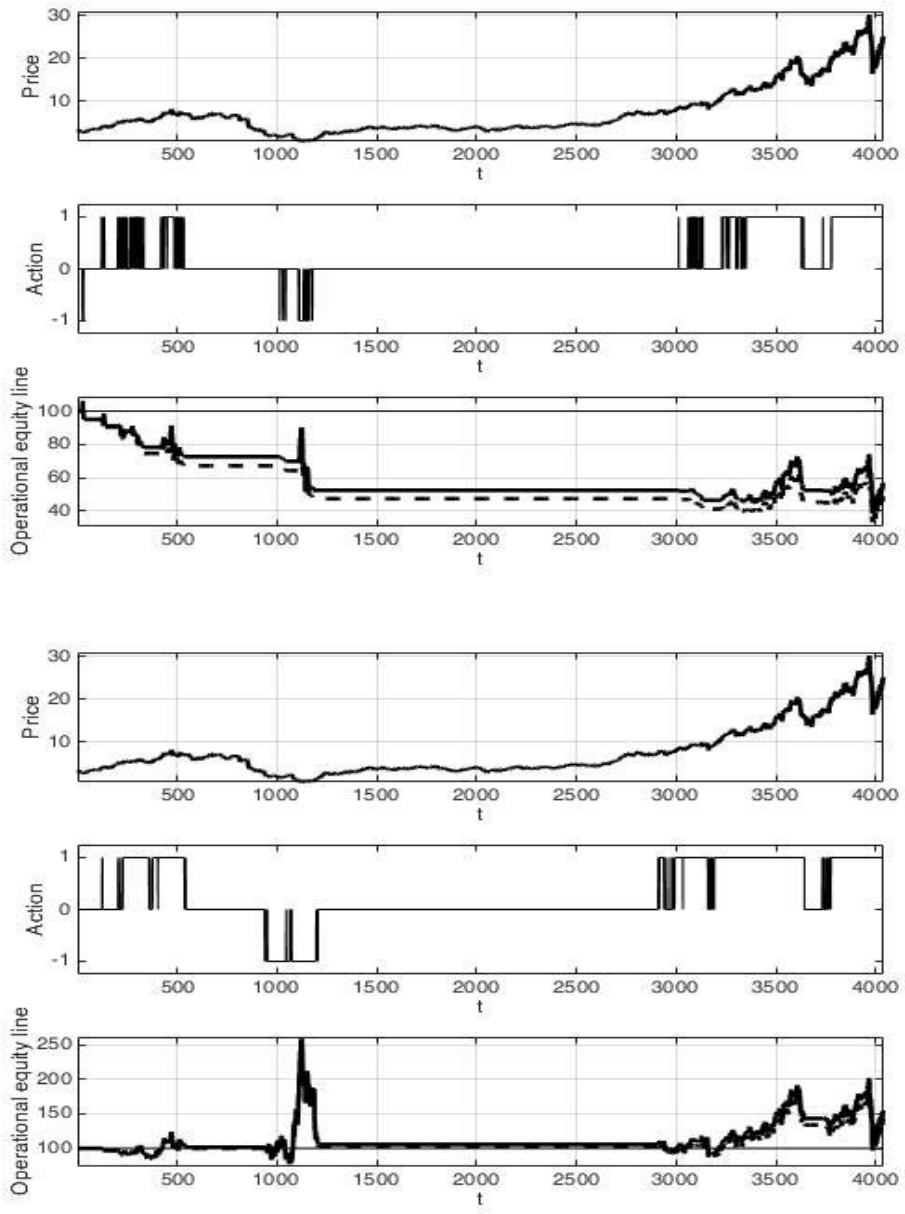

Figura 5.1: QL con N=1, L=5, $\varepsilon = 5\%$. In alto Burke ratio, in basso Sharpe ratio. Fonte: Matlab.

potrebbe rendere Burke meno efficiente e meno sensibile alle piccole oscillazioni rispetto a Sharpe. Si consideri anche che il caso di specie considera $L = 5$ e $N = 1$, quindi pochi rendimenti passati per valutare la volatilità e pochi elementi nel vettore della struttura degli stati che porta a maggiore instabilità nel segnale.

Passando ad analizzare il caso di Sortino con $N = 1$ $\varepsilon = 15\%$, si vedano i grafici in Figura 5.2, dove viene rappresentato in alto $L = 5$, in basso $L = 22$. Con questo confronto si vuole indagare la causa della differenza di rendimenti che

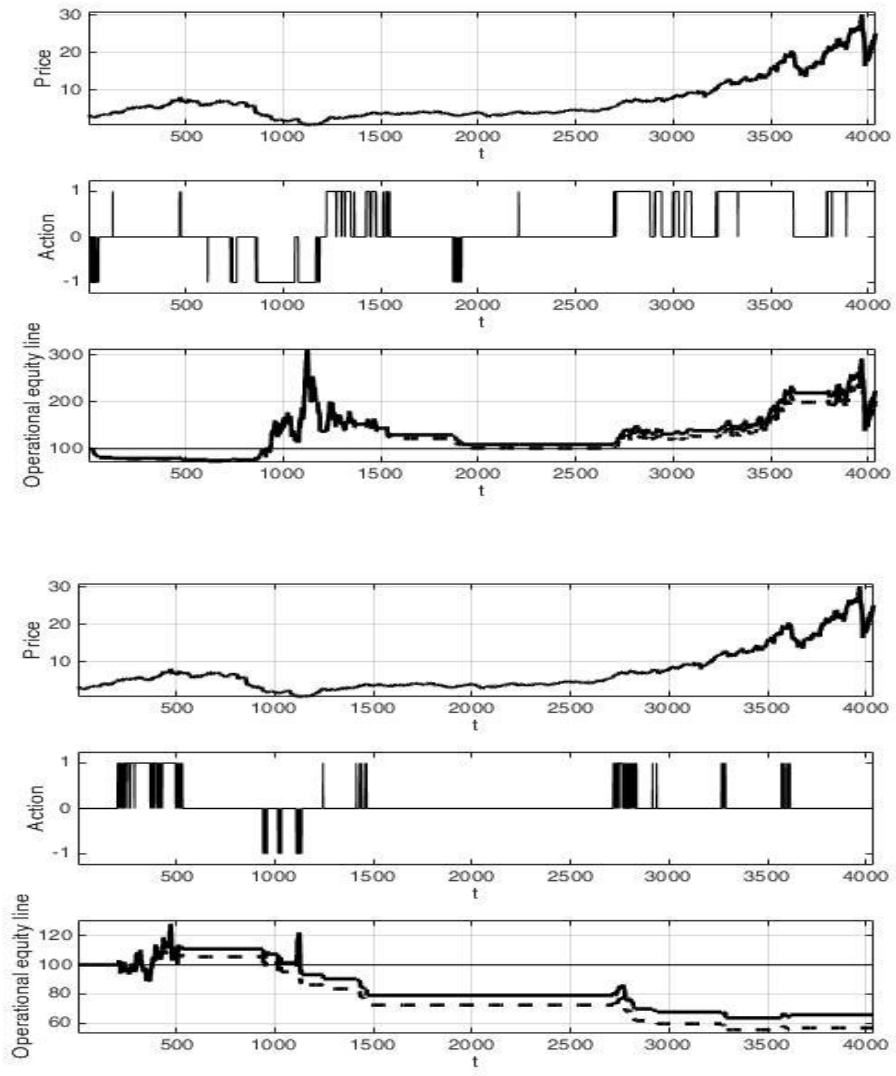

Figura 5.2: Sortino ratio, QL con N=1, $\varepsilon = 15\%$. In alto L=5, in basso $L = 22$. Fonte: Matlab

insorge tra i due valori di L nel caso di Sortino ratio. Si nota come la sola differenza del parametro L porti ad una notevole diversità nel secondo e terzo pannello dei due grafici. Sembra che nel caso $L = 5$ l'algoritmo sia più reattivo nel primo periodo e pronto a reagire alle oscillazioni dei prezzi, che anche se piccole e senza un chiaro trend. Al contrario con $L = 22$ l'algoritmo non riesce a tradurre le oscillazioni in segnali operativi efficienti e causa una serie di perdite che mantiene e peggiora nel corso dell'investimento. Questa differenza di valori di L provoca una maggiore o minore sensibilità alla volatilità, che nel caso di $L = 5$ si riduce, facendo sì che l'algoritmo reagisca basandosi solo sui pochi precedenti rendimenti che ha realizzato. Al contrario, con $L = 22$ il Sortino ratio

considera gli ultimi 22 giorni di trading che, nel caso di una situazione dei prezzi senza trend, fa sì che i segnali perdano di potere informativo. Un altro effetto di questa diversa reazione si vede nel numero di operazioni che vengono compiute, che è maggiore in $L = 5$ e minore in $L = 22$. Infine, in generale sono leggermente migliori le performance per $N = 5$ in tutti gli indici.

Nelle prossime Tabelle 5.3 e 5.4 si presentano i risultati che si sono ottenuti utilizzando l'algoritmo SARSA. Si nota che le performance di SARSA peggiorano per tutti i ratio rispetto ai risultati ottenuti con QL. La Tabella 5.3 presenta i risultati di SARSA con $N = 1$, mentre la tabella successiva presenta i risultati per $N = 5$. Dalla prima analizzando Sharpe si può apprezzare che rispetto al caso QL (in particolare rispetto alla Tabella 5.1) ci sono dei rendimenti negativi, mentre quelli positivi sono decisamente di grandezza minore. In particolare con Sharpe si ottiene il rendimento più basso della tabella quando $L = 22$ e $\varepsilon = 15\%$, pari a -4.033% cioè ad un capitale finale netto di $56{,}641€$.

N=1	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	L	ε (%)	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	103,359	**101,660**	0,001	**0,000**	0,206	**0,103**	93,287	**93,138**	**1,4**
		10	103,006	**99,216**	0,001	**0,000**	0,185	**-0,049**	97,250	**10,107**	**3,1**
		15	149,347	**127,897**	0,010	**0,006**	2,535	**1,548**	98,861	**97,919**	**12,9**
	22	5	93,267	**90,239**	-0,002	**-0,003**	-0,434	**-0,639**	16,968	**16,027**	**2,7**
		10	114,798	**101,861**	0,003	**0,000**	0,865	**0,115**	36,116	**31,930**	**9,9**
		15	56,641	**51,714**	-0,014	**-0,016**	-3,486	**-4,033**	17,414	**16,473**	**7,6**
BR	5	5	86,604	**84,544**	-0,004	**-0,004**	-0,894	**-1,043**	0,694	**0,694**	**2,0**
		10	84,353	**76,807**	-0,004	**-0,007**	-1,057	**-1,634**	0,892	**0,669**	**7,8**
		15	82,106	**73,085**	-0,005	**-0,008**	-1,223	**-1,938**	6,713	**5,004**	**9,7**
	22	5	59,970	**59,076**	-0,013	**-0,013**	-3,141	**-3,232**	9,983	**9,240**	**1,2**
		10	79,122	**78,181**	-0,006	**-0,006**	-1,451	**-1,525**	16,151	**15,730**	**1,0**
		15	101,450	**96,552**	0,000	**-0,001**	0,090	**-0,219**	95,962	**23,929**	**4,1**
SOR	5	5	144,629	**133,655**	0,009	**0,007**	2,330	**1,827**	13,401	**11,147**	**6,6**
		10	183,609	**163,933**	0,015	**0,012**	3,866	**3,134**	20,857	**17,736**	**9,4**
		15	122,627	**109,148**	0,005	**0,002**	1,281	**0,548**	16,423	**12,063**	**9,7**
	22	5	68,731	**68,423**	-0,009	**-0,009**	-2,314	**-2,341**	21,130	**21,130**	**0,4**
		10	73,795	**71,823**	-0,008	**-0,008**	-1,879	**-2,045**	8,967	**8,571**	**2,2**
		15	94,321	**88,290**	-0,001	**-0,003**	-0,364	**-0,774**	14,392	**12,014**	**5,5**

Tabella 5.3: Amplifon S.p.A. per SARSA con N=1

N=5	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	L	ε (%)	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	84,996	**82,973**	-0,004	**-0,005**	-1,011	**-1,160**	5,877	**5,529**	**2,0**
		10	104,717	**103,004**	0,001	**0,001**	0,288	**0,185**	95,958	**95,810**	**1,4**
		15	98,755	**90,694**	0,000	**-0,002**	-0,078	**-0,608**	26,432	**26,060**	**7,1**
	22	5	131,087	**116,799**	0,007	**0,004**	1,706	**0,975**	94,843	**84,354**	**9,6**
		10	97,971	**93,004**	-0,001	**-0,002**	-0,128	**-0,452**	30,672	**30,126**	**4,3**
		15	99,931	**81,788**	0,000	**-0,005**	-0,004	**-1,248**	24,969	**12,100**	**16,7**
BR	5	5	105,295	**99,924**	0,001	**0,000**	0,323	**-0,005**	89,388	**0,645**	**4,4**
		10	76,259	**70,744**	-0,007	**-0,009**	-1,679	**-2,139**	0,570	**0,570**	**6,2**
		15	121,197	**110,853**	0,005	**0,003**	1,209	**0,646**	20,233	**16,464**	**7,4**
	22	5	55,994	**54,991**	-0,014	**-0,015**	-3,559	**-3,668**	6,819	**5,901**	**1,5**
		10	88,711	**86,601**	-0,003	**-0,004**	-0,746	**-0,895**	11,827	**11,208**	**2,0**
		15	108,559	**104,261**	0,002	**0,001**	0,514	**0,261**	95,115	**94,719**	**3,4**
SOR	5	5	165,683	**154,964**	0,013	**0,011**	3,205	**2,775**	17,927	**15,125**	**5,6**
		10	183,732	**174,474**	0,015	**0,014**	3,874	**3,539**	24,151	**23,382**	**4,3**
		15	175,032	**166,212**	0,014	**0,013**	3,560	**3,226**	93,628	**90,032**	**4,3**
	22	5	70,938	**69,039**	-0,009	**-0,009**	-2,123	**-2,288**	20,828	**20,754**	**2,2**
		10	125,499	**119,434**	0,006	**0,004**	1,429	**1,116**	99,826	**99,826**	**4,1**
		15	124,775	**114,202**	0,005	**0,003**	1,393	**0,833**	98,959	**98,041**	**7,4**

Tabella 5.4: Amplifon S.p.A. per SARSA con N=5

Guardando i risultati ottenuti con Burke si nota che sono tutti negativi per ogni ε e ogni L. In generale non sembra ci siano evidenze al variare di ε o di L. A questi risultati si associano anche percentuali over 100 decisamente basse, nessuna delle quali supera il 16%; questo significa che non solo gli investimenti terminano (in media) con valori negativi, ma anche durante la maggior parte del tempo di investimento la performance è negativa. Con Sortino la situazione cambia leggermente, poiché si ottengono rendimenti positivi per $L = 5$, negativi per $L = 22$. Tuttavia, guardando le percentuali over 100 a cui si associano i valori, anche queste risultano molto basse, ed anche per $L = 5$ non superano il valore di 18%. Questo può suggerire che nonostante siano terminati con rendimenti positivi, in generale l'investimento non abbia avuto performance positive, tanto che se si fosse interrotto in un qualsiasi momento ci sarebbe stato più del 82% di possibilità di avere un rendimento negativo. Quindi con SARSA per $N = 1$, i risultati sono in generale peggiori di QL per tutte le funzioni di reward utilizzate. Resta il fatto che Sharpe performa meglio di Burke e Sortino, mentre Burke ottiene le performance

peggiori. Guardando l'ultima tabella con i dati di SARSA con $N = 5$ per Sharpe e Burke ratio alcuni valori migliorano altri peggiorano, mentre con Sortino ratio migliorano i valori di tutte le combinazioni. Quindi tra $N = 1$ e $N = 5$ per Sharpe e Burke ratio non ci sono evidenze, mentre sembra che Sortino tragga vantaggio dall'uso di più elementi nel vettore s_t soprattutto per $L = 22$ dove ottiene rendimenti positivi quando invece per $N = 1$ erano tutti negativi. Si può ipotizzare che la presenza di più elementi nel vettore stato ad attenuare la difficoltà che Sortino ha mostrato con $L = 22$ in periodi di prezzi senza trend. Lo stesso vale per le percentuali over 100, che confermano quanto congetturato.

Si vedano i seguenti grafici in Figura 5.3 a tre pannelli che confrontano l'output di Sortino ratio con SARSA $L = 22$ e $\varepsilon = 15\%$ per accostare le performance tra il caso $N = 1$ e $N = 5$.

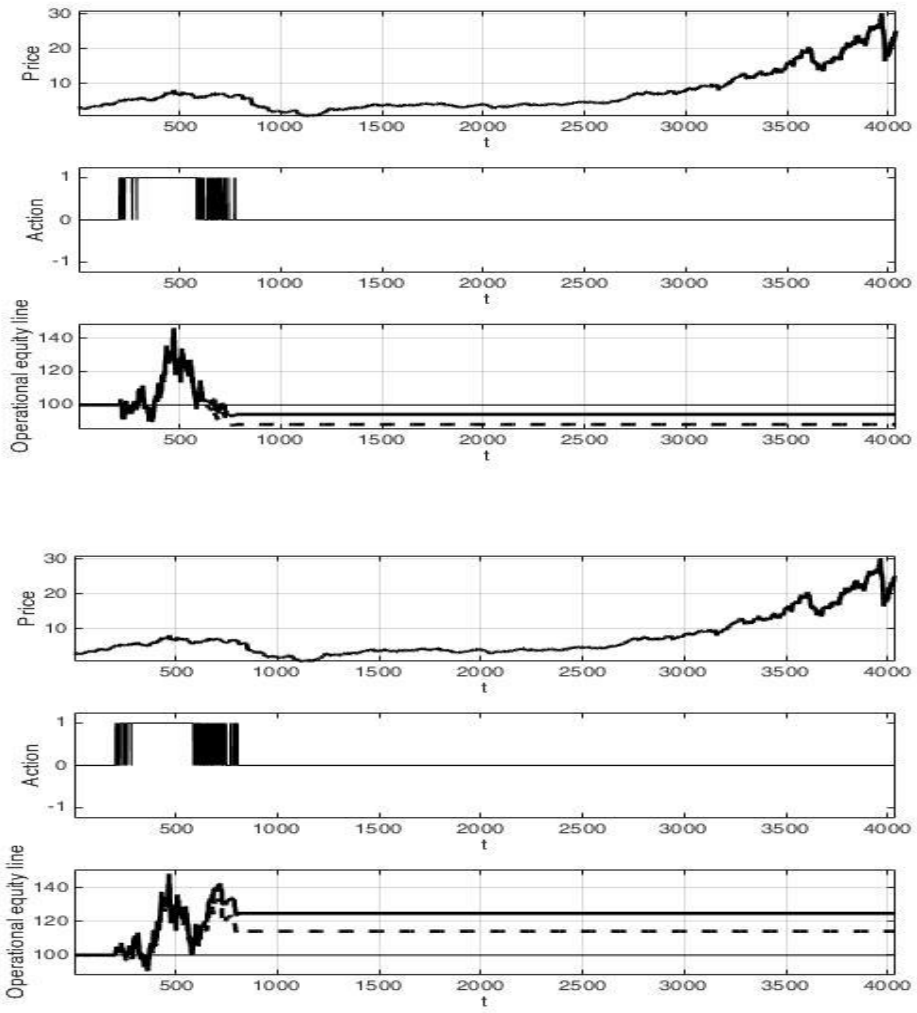

Figura 5.3: Sortino ratio, SARSA con $L = 22$, $\varepsilon = 15\%$. In alto $N = 1$, in basso $N = 5$. Fonte: Matlab

Dai pannelli delle azioni compiute non si evidenziano rilevanti differenze in termini di azioni intraprese, inoltre dalle tabelle risulta che il numero di operazioni compiute all'anno sia pressoché simile. Tuttavia, le operazioni nel primo periodo determinano un'*equity line* molto diversa tra i due casi dopo l'istante $t = 500$, in corrispondenza del quale inizia un periodo di oscillazioni dove la linea dei prezzi è particolarmente piatta. Come ipotizzato sembra che la presenza di più elementi che descrivono il vettore di stato in questo caso riescano a determinare un rendimento positivo, a differenza del caso in cui $N = 1$. Da un confronto tra questa figura e la precedente Figura 5.2 si nota la notevole differenza di performance tra QL e SARSA, dove il primo si percepisce più reattivo con un discreto numero di azioni compiute all'anno, mentre SARSA appare rigido, con poche azioni (da circa $t = 800$ resta sempre fuori dal mercato).

Si può quindi concludere che le performance relative ad Amplifon S.p.A. con SARSA in generale peggiorano rispetto a QL. In QL però non ci sono differenze di performance degli indici al variare di N a differenza di quanto avviene con SARSA dove in particolare Sortino migliora con $N = 5$. In generale considerando tutte e quattro le tabelle di output si può apprezzare che il valore di performance maggiore ottenuto sia il valore di Sharpe con l'algoritmo QL per $N = 1, L = 22$ ed $\varepsilon = 10\%$ pari ad un rendimento annuo netto di $8{,}868\%$ cioè un capitale finale netto di $390{,}06€$. Dalla stessa tabella in Burke con $L = 5$ ed $\varepsilon = 5$ si ricava anche il valore di performance peggiore, pari a $-4{,}398\%$ cioè un capitale finale netto di $48{,}648€$ che consiste in meno della metà del capitale inizialmente investito. A questo caso si associa anche una pessima percentuale over 100 pari a $0{,}694\%$. Non ci sono evidenze particolari che suggeriscano che gli indici performino meglio o peggio al variare del valore di *exploration*. per gli altri indici. Sembra che Sharpe sia leggermente più performante per maggiori valori di ε. In generale risulta quindi che Sharpe sembra la misura di performance migliore, davanti alla misura di Sortino mentre Burke risulta il meno performante.

Si osservino i seguenti tre grafici (Figura $5.4, 5.5, 5.6$) che rappresentano i casi di QL con $N = 1, L = 5, \varepsilon = 5\%$, rispettivamente per Sharpe, Burke e Sortino. In questo modo si cerca di confrontare visivamente l'andamento dell'investimento in termini di azioni e performance ottenute così da vedere se ci

siano particolari evidenze al pari dei parametri e dell'algoritmo, eliminando il più possibile la componente random dell'*exploration*. In particolare questo è il caso in cui si ha l'unico rendimento negativo della prima tabella, risultato della misura di Burke ratio. In questi grafici è interessante notare come Sharpe e Sortino abbiano andamenti molto simili guardando il pannello delle azioni intraprese sia guardando l'*equity line* risultate, mentre per Burke l'andamento dell'*equity line* è molto diverso dagli altri. È evidente che all'inizio dell'investimento, anche se i pannelli delle azioni intraprese mostrano in tutti e tre i casi diverse posizioni *long* nello stesso periodo, il risultato per Burke ottiene rendimenti negativi che continuano a peggiorare all'avanzare dei giorni. Per Sharpe e Sortino invece si vede un periodo negativo che prosegue fino a poco prima del 1000-esimo giorno di trading, dove poi diventa positivo. Dopo un periodo di uscita dal mercato (cioè $a_t = 0$) presente per tutte le misure, intorno al giorno 3000 di trading l'andamento dei prezzi assume un trend crescente che viene tradotto dagli algoritmi con alcune azioni di acquisto che risultano in una buona performance per Sharpe e Sortino che passano da un valore di investimento negativo (anche se vicino al valore di 100€) ad un valore positivo. Interessante l'andamento dell'*equity line* di Burke che rileva allo stesso modo i segnali operativi come per Sharpe e Sortino, tuttavia parte da un valore di *equity line* basso a causa del quale, anche migliorando, non ottiene un rendimento positivo. È possibile quindi concludere che in questo esempio Burke non sia stato in grado di interpretare i segnali operativi all'inizio del periodo, ottenendo quindi performance negative per il resto del periodo di investimento. È possibile che Burke non sia in grado di ottenere dei segnali tali da entrare nel mercato quando il prezzo non ha (o non ha ancora) una tendenza molto chiara. Se si osserva l'istante in cui verso la fine dell'investimento i tre indici reagiscono al trend di crescita dei prezzi, si nota che Sortino è il primo che entra nel mercato in posizione di acquisto in circa $t = 2700$, Sharpe è il secondo entrando nel mercato circa in $t = 2900$ mentre Burke è l'ultimo ed assume una posizione di acquisto solo in $t = 3000$. In più, Burke è l'indice che resiste per più tempo fuori dal mercato rispetto agli altri.

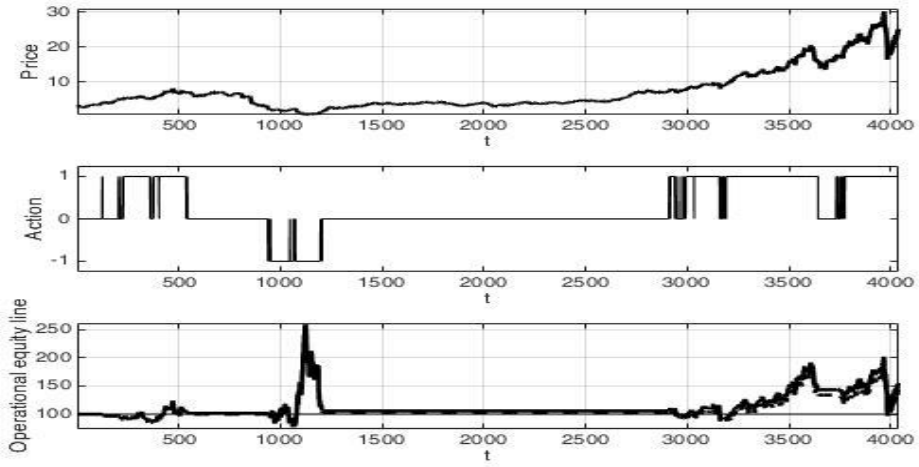

Figura 5.4: Amplifon S.p.A., QL, Sharpe ratio con N=1, L=5, $\varepsilon = 5\%$. Fonte: Matlab.

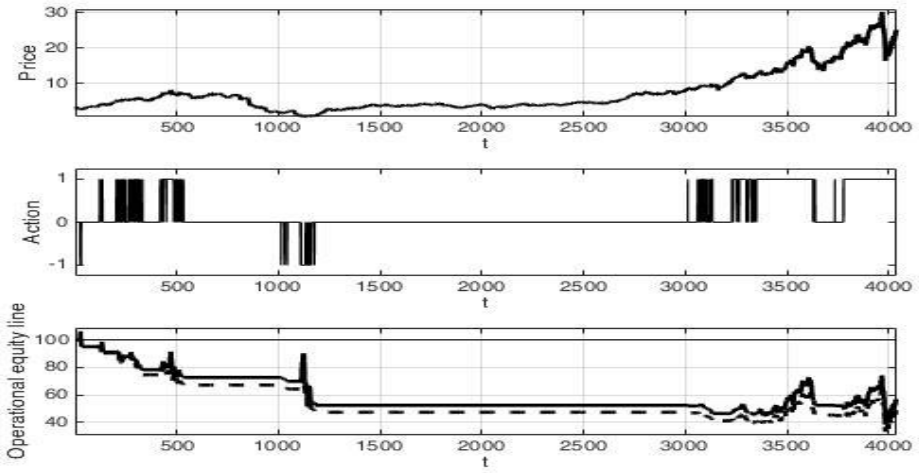

Figura 5.5: Amplifon S.p.A., QL, Burke ratio con N=1, L=5, $\varepsilon = 5\%$. Fonte: Matlab.

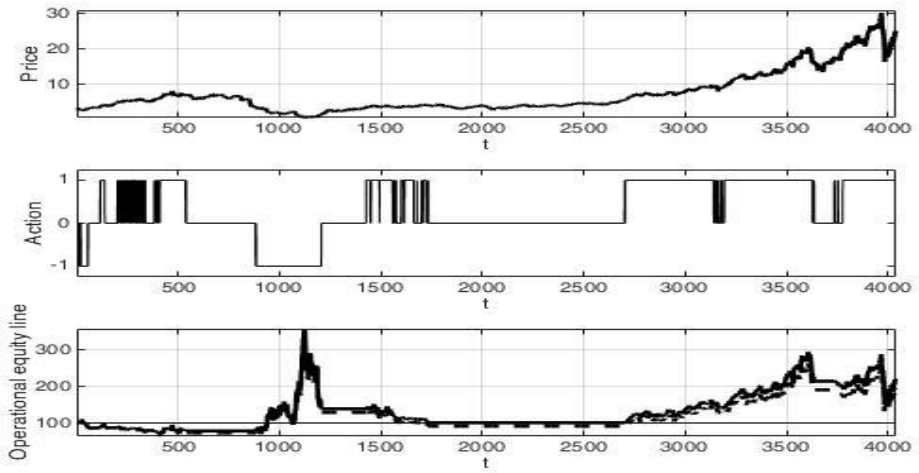

Figura 5.6: Amplifon S.p.A., QL, Sortino ratio con N=1, L=5, $\varepsilon = 5\%$. Fonte: Matlab

Si prendono ora di esempio i grafici nel caso in cui tutti e tre i ratio performano in modo simile, ovvero quando si utilizza l'algoritmo QL con $N = 1, L = 5, \varepsilon = 10\%$. A differenza di prima (Figura 5.4, 5.5, 5.6), tutti e tre i grafici hanno un simile andamento delle *equity lines* che risultano dalle azioni intraprese. Il maggior valore di ε si nota dal fatto che aumenta il numero di azioni che vengono compiute nel periodo dove nel caso di $\varepsilon = 5\%$ rimangono tutti fuori dal mercato. Un'altra osservazione risulta guardando il primo periodo di investimento. Mentre Sharpe rimane fuori dal mercato, sia Burke che Sortino assumono una posizione di vendita del titolo che li porta a una perdita che rimane per diverso

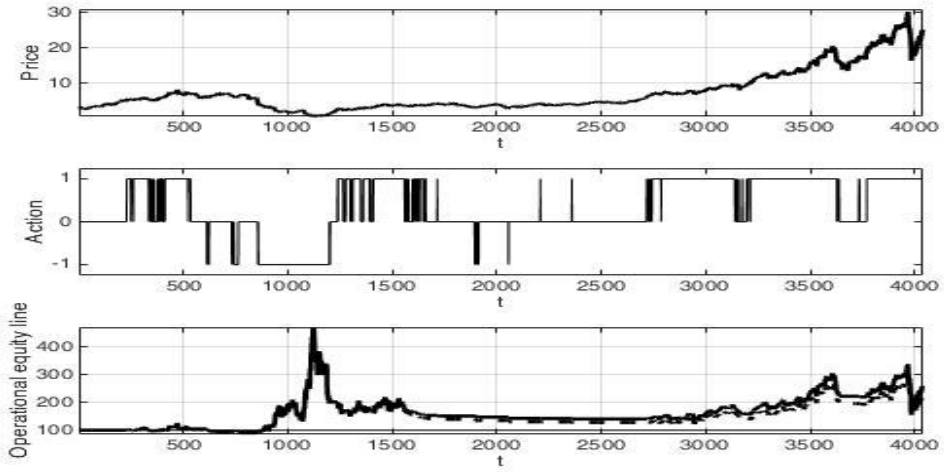

Figura 5.7: Amplifon S.p.A., QL, Sharpe ratio con N=1, L=5, $\varepsilon = 10\%$. Fonte: Matlab.

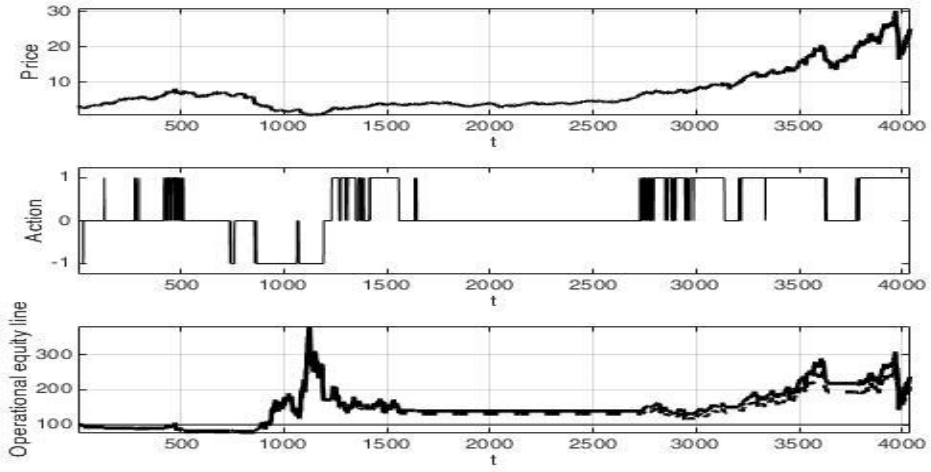

Figura 5.8: Amplifon S.p.A., QL, Burke ratio con N=1, L=5, $\varepsilon = 10\%$. Fonte: Matlab.

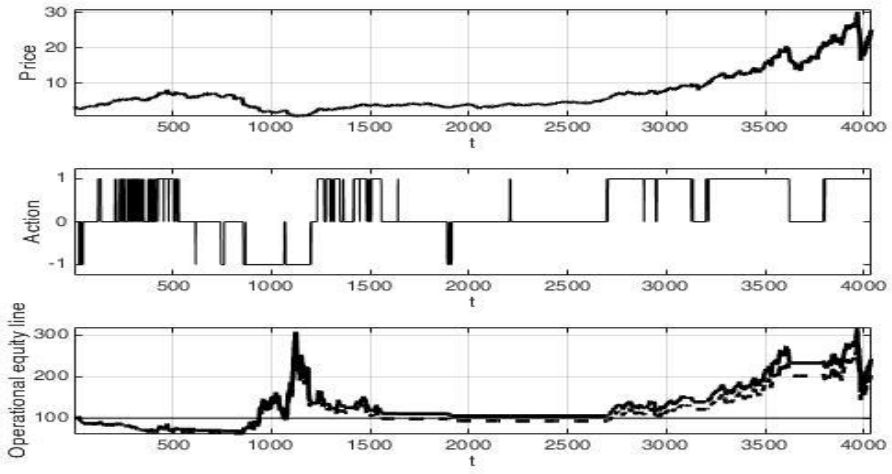

Figura 5.9: Amplifon S.p.A., QL, Sortino ratio con N=1, L=5, $\varepsilon = 10\%$. Fonte: Matlab.

tempo, circa fino a poco prima dei 1000-esimo giorno di trading dove tutti e tre assumono una posizione short che li porta ad ottenere performance positive.

Un ultimo esempio che vale la pena di riportare è il caso delle performance ottenute con l'algoritmo SARSA per gli stessi parametri appena analizzati, dunque con $N = 1, L = 5, \varepsilon = 10\%$. Si riportano in ordine il caso di Sharpe (Figura 5.10), Burke (Figura 5.11) e Sortino (Figura 5.12). Si nota come i pannelli siano molto diversi dal caso QL. In primo luogo il periodo di uscita dal mercato è costante e molto più lungo in tutti e tre i casi. Questo potrebbe indicare che nel caso di SARSA siano necessari segnali "più forti" perché l'algoritmo prenda posizione. Questo è evidente in particolare nel caso di Sharpe, dove l'algoritmo da poco dopo il giorno $t = 500$ rimane fuori dal mercato fino alla fine dell'investimento, nonostante ci sia una tendenza positiva dei prezzi del titolo al termine del periodo. Burke e soprattutto Sortino invece compiono più azioni, con la differenza che Sortino ottiene dei guadagni anche molto positivi, mentre Burke al contrario ottiene rendimenti negativi. Sembra quindi da questo caso che SARSA sia un algoritmo più rigido, meno sensibile alle variazioni di prezzo rispetto a QL, e che sia meno performante visto che con QL si ottengono rendimenti in generale maggiori. In più, Sortino appare più abile rispetto a Burke quando si utilizza SARSA come algoritmo.

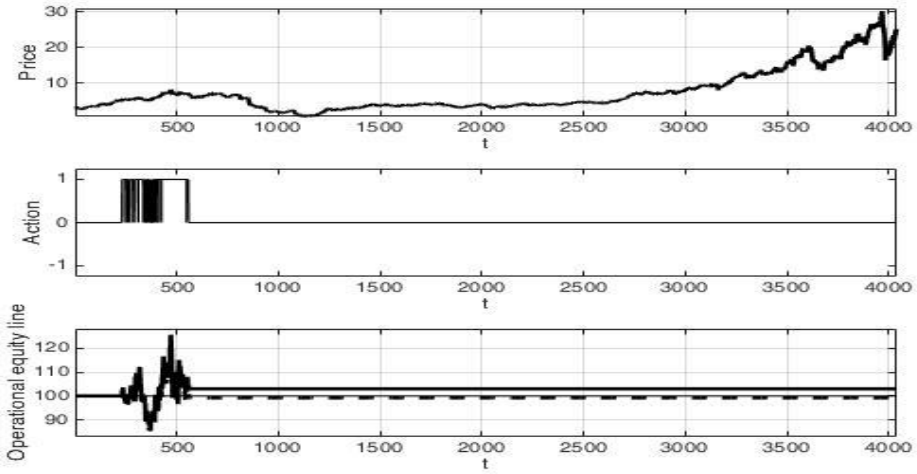

Figura 5.10: Amplifon S.p.A., Sharpe ratio SARSA con N=1, L=5, $\varepsilon = 10\%$. Fonte: Matlab.

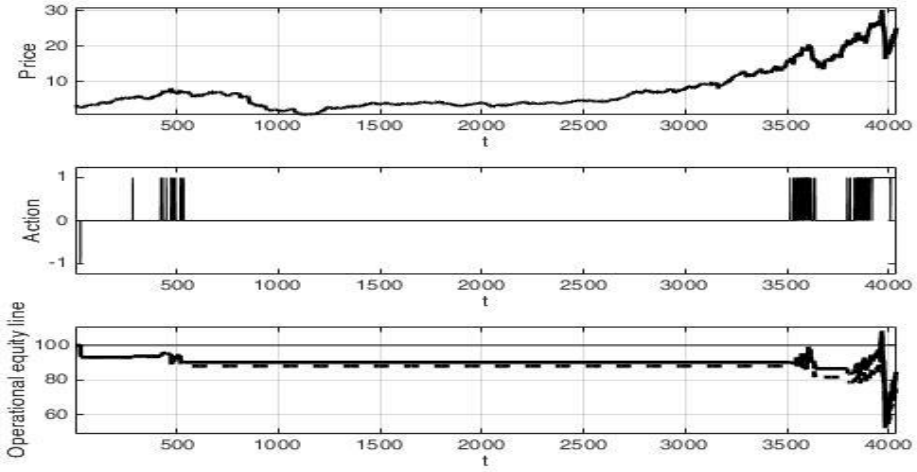

Figura 5.11: Amplifon S.p.A., Burke ratio SARSA con N=1, L=5, $\varepsilon = 10\%$. Fonte: Matlab.

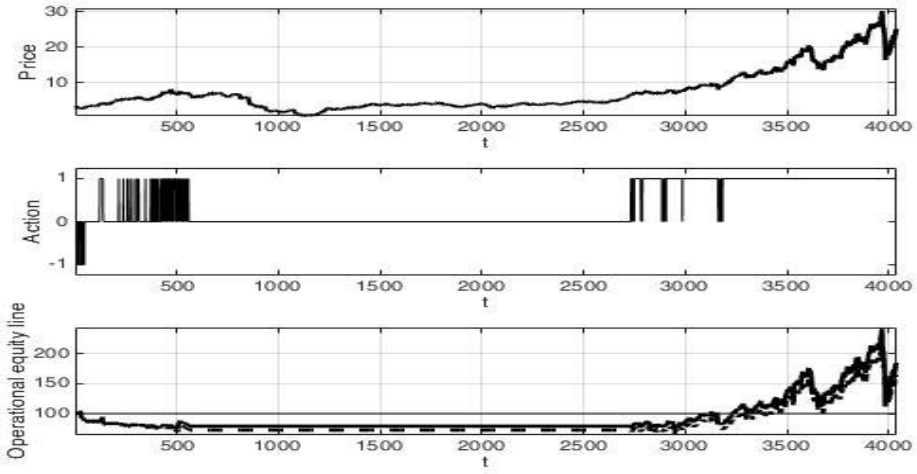

Figura 5.12: Amplifon S.p.A., Sortino ratio SARSA con N=1, L=5, $\varepsilon = 10\%$. Fonte: Matlab.

Un utile confronto tra gli algoritmi può risultare dai grafici con tutte le $k = 500$ iterazioni che gli algoritmi hanno compiuto per ottenere questi risultati e i grafici con la linea media di queste iterazioni (che corrisponde quindi all'*equity line* finale). Si vedano i prossimi grafici (Figura 5.13) per Amplifon S.p.A. con $N = 1, L = 5, \varepsilon = 10\%$ e utilizzando lo Sharpe ratio come funzione di reward. È chiaro come nel caso di SARSA le *equity lines* siano molto più diffuse, con traiettorie anche molto diverse tra loro, mentre nel caso di QL nella maggior parte siano concentrate in un range di valori di investimento più contenuti. Risulta quindi dalle medie di queste *equity lines* che all'aumentare del periodo di tempo t di investimento, i due algoritmi ottengano risultati diversi. La media delle *equity lines* più diffuse in SARSA risulta in un'*equity line* con valori più bassi di quelli ottenuti da QL, in particolare per il periodo successivo a $t = 1500$.

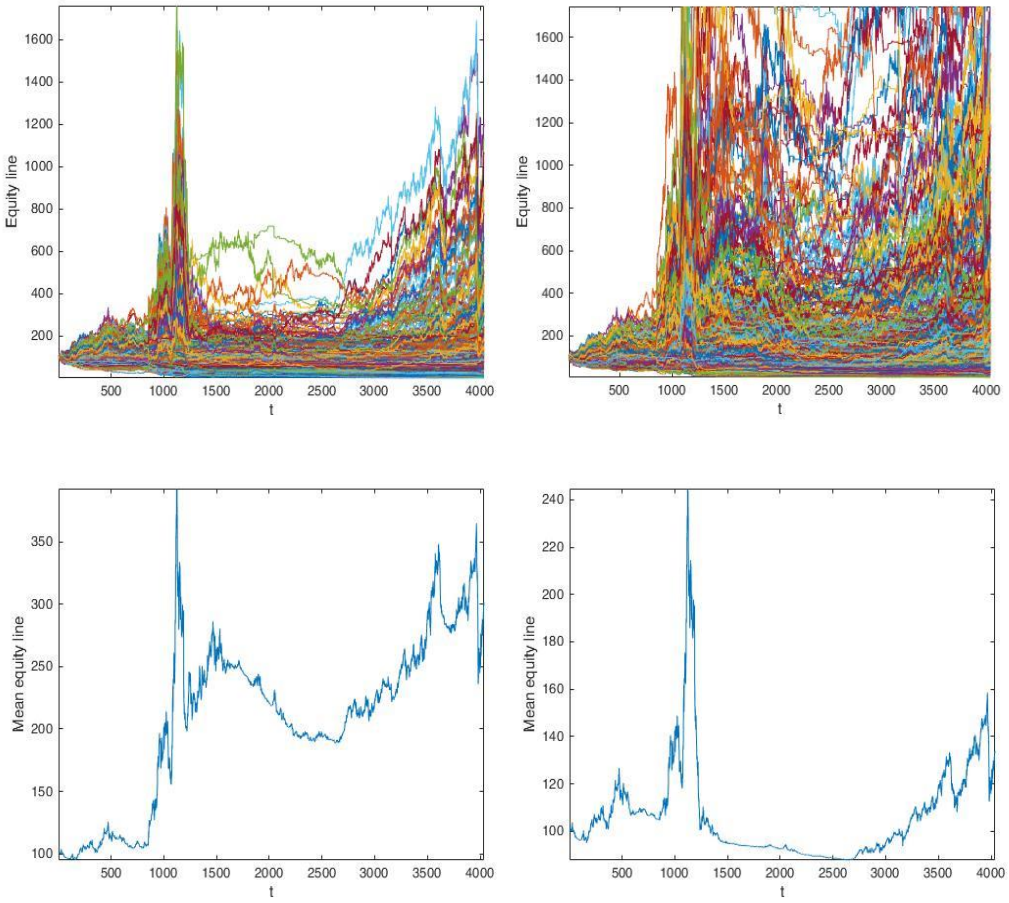

Figura 5.13: Amplifon S.p.A., Sharpe ratio SARSA con $N = 1, L = 5, \varepsilon = 10\%$. In alto: equity lines risultanti per QL (s sinistra) e SARSA (a destra). In basso: equity line media per QL (a sinistra), SARSA (a destra). Fonte: Matlab

Per riassumere quanto visto, il caso del titolo Amplifon S.p.A. mostra una differenza notevole tra le performance di QL e SARSA. SARSA appare un algoritmo meno sensibile alle variazioni di prezzo e che porta a rendimenti inferiori a quelli ottenuti da QL, in generale per tutti i tipi di funzioni di *reward*. Tra le misure di performance con QL, Sharpe risulta essere il più performante con capitale finale, rendimenti e percentuali over 100 migliori rispetto alle altre due misure. Burke appare il meno performante dei tre, con rendimenti più bassi e maggiore rigidità alle variazioni dei prezzi che lo rende meno ricettivo rispetto ai segnali, probabilmente a causa dell'andamento dei prezzi del titolo Amplifon S.p.A. che, soprattutto nel primo periodo, è privo di trend. Sortino invece con $L = 22$ non riesce a tradurre le oscillazioni in segnali operativi efficienti poiché considerando gli ultimi 22 giorni di trading in una situazione dei prezzi senza trend fa sì che i segnali perdano di potere informativo. Sortino considera solo gli scostamenti negativi ma in modo più vicino a Sharpe, facendo sì che le sue performance siano più simili a Sharpe rispetto a quelle ottenute con Burke, che invece considera solo le variazioni negative in termini di *drawdown* e quindi in modo meno sensibile alle piccole oscillazioni rispetto a Sharpe.

5.1.2 Azimut S.p.A.

Passando al titolo successivo, si riportano in Tabella 5.5 i risultati relativi ad Azimut S.p.A. per QL con $N = 1$. Nel caso di Azimut S.p.A. dalla prima tabella si osserva che Sharpe ottiene rendimenti negativi per ε piccoli, positivi per ε maggiori. Diverso invece con Burke dove non sembra esserci una regola: in $L = 5$ i rendimenti sono migliori quando ε è piccolo, il contrario vale quando $L = 22$. Unico rendimento negativo per $L = 5$ $\varepsilon = 5\%$. Simile il risultato per Sortino: ottiene rendimenti positivi a parte nel caso $L = 5$ $\varepsilon = 5\%$, mentre i rendimenti maggiori li ottiene per $\varepsilon = 10\%$ per ogni L. Ottimi in generale i valori di percentuali over 100 che sono alti anche in Sharpe dove ottiene più rendimenti, mentre per Burke e Sortino sono prossimi al 100% (in due casi anche uguale a 100%). Da notare che la percentuale over 100 che ottiene Sharpe in

corrispondenza dei risultati negativi è comunque migliore di quella di Burke e Sortino quando ottengono

N=1	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	102,165	**93,787**	0,001	**-0,002**	0,134	**-0,400**	99,331	**71,291**	7,1
		10	99,181	**89,364**	0,000	**-0,003**	-0,051	**-0,699**	97,919	**89,398**	8,7
		15	146,583	**132,484**	0,009	**0,007**	2,416	**1,771**	100,000	**99,851**	8,4
	22	5	70,201	**62,497**	-0,009	**-0,012**	-2,184	**-2,892**	77,582	**72,034**	9,7
		10	126,332	**114,676**	0,006	**0,003**	1,470	**0,859**	99,133	**98,737**	8,1
		15	185,103	**164,547**	0,015	**0,012**	3,918	**3,158**	99,653	**99,628**	9,8
BR	5	5	205,170	**200,000**	0,018	**0,017**	4,588	**4,422**	99,926	**99,430**	2,1
		10	158,218	**144,917**	0,011	**0,009**	2,905	**2,343**	99,950	**99,950**	7,3
		15	103,909	**90,979**	0,001	**-0,002**	0,240	**-0,588**	99,975	**98,885**	11,0
	22	5	125,944	**113,892**	0,006	**0,003**	1,450	**0,815**	95,640	**95,467**	8,4
		10	141,305	**127,425**	0,009	**0,006**	2,182	**1,524**	99,009	**98,836**	8,6
		15	164,973	**150,072**	0,012	**0,010**	3,174	**2,566**	96,185	**95,863**	7,9
SOR	5	5	98,846	**91,342**	0,000	**-0,002**	-0,072	**-0,564**	98,563	**98,167**	6,6
		10	168,942	**154,317**	0,013	**0,011**	3,328	**2,745**	99,183	**99,133**	7,6
		15	157,028	**135,757**	0,011	**0,008**	2,857	**1,927**	99,430	**99,356**	12,1
	22	5	127,807	**112,819**	0,006	**0,003**	1,543	**0,756**	81,570	**77,954**	10,4
		10	207,018	**175,042**	0,018	**0,014**	4,647	**3,557**	99,950	**99,926**	14,0
		15	148,797	**124,454**	0,010	**0,005**	2,512	**1,375**	99,207	**98,984**	14,9

Tabella 5.5: Azimut S.p.A. per QL con N=1

N=5	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	82,720	**73,910**	-0,005	**-0,007**	-1,178	**-1,871**	63,427	**54,922**	9,4
		10	116,166	**106,064**	0,004	**0,001**	0,941	**0,369**	99,231	**98,562**	7,6
		15	170,886	**152,591**	0,013	**0,010**	3,405	**2,676**	99,455	**99,455**	9,4
	22	5	87,317	**82,907**	-0,003	**-0,005**	-0,844	**-1,164**	91,322	**88,743**	4,3
		10	121,245	**109,422**	0,005	**0,002**	1,211	**0,564**	99,702	**99,504**	8,6
		15	179,030	**159,112**	0,014	**0,012**	3,706	**2,945**	99,752	**99,678**	9,8
BR	5	5	101,991	**94,204**	0,000	**-0,001**	0,123	**-0,372**	99,628	**37,292**	6,6
		10	184,263	**164,631**	0,015	**0,012**	3,893	**3,164**	99,653	**99,603**	9,4
		15	195,544	**175,287**	0,017	**0,014**	4,279	**3,569**	99,851	**99,851**	9,1
	22	5	265,649	**241,372**	0,024	**0,022**	6,295	**5,660**	98,587	**98,562**	8,0
		10	353,175	**317,477**	0,031	**0,029**	8,203	**7,485**	99,083	**99,033**	8,9
		15	335,858	**291,730**	0,030	**0,027**	7,864	**6,919**	100,000	**100,000**	11,7
SOR	5	5	60,732	**54,797**	-0,012	**-0,015**	-3,068	**-3,689**	71,733	**67,444**	8,6
		10	157,480	**143,147**	0,011	**0,009**	2,878	**2,267**	99,529	**99,455**	7,9
		15	314,936	**276,393**	0,028	**0,025**	7,431	**6,559**	99,554	**99,430**	10,9
	22	5	259,062	**236,457**	0,024	**0,021**	6,128	**5,525**	99,603	**99,603**	7,6
		10	225,877	**209,560**	0,020	**0,018**	5,223	**4,731**	99,950	**99,950**	6,2
		15	140,859	**130,856**	0,008	**0,007**	2,164	**1,695**	100,000	**100,000**	6,1

Tabella 5.6: Azimut S.p.A. per QL con N=5

rendimenti negativi. La migliore performance con QL e $N = 1$ si ottiene con Burke quando $L = 5$ e $\varepsilon = 5\%$ dove raggiunge un rendimento annuo netto di $4{,}422\%$ pari a $200{,}000€$ al netto dei costi.

Spostandosi sulla Tabella 5.6 dove sono rappresentati i risultati per QL con $N = 5$ si nota che ci sono miglioramenti nella maggior parte delle combinazioni, soprattutto per $L = 22$ con Burke e Sortino. Le percentuali over 100 restano in linea quelle della tabella $N = 1$. In generale Sharpe è il ratio che performa peggio tra i tre in termini di rendimenti, in particolare quando $\varepsilon = 5\%$. Tuttavia le percentuali over 100 risultano prossime al 100% o comunque alte, anche quando ottiene risultati negativi: anche se gli investimenti si concludono con delle perdite, non significa che la performance sia stata negativa durante tutto il periodo. Sortino e, in particolare, Burke performano meglio di Sharpe, in particolare quando $N = 5$. Si nota infatti che entrambi i ratio migliorano i rendimenti per $L = 22$ quando passano da $N = 1$ a $N = 5$. Questo comportamento sembra simile all'analisi del titolo precedente quando la mancanza di un trend confonde i ratio che basano le valutazioni su un intervallo di giorni passati di trading maggiore (cioè $L = 22$). Si nota anche che tutte le funzioni di reward ottengono le performance peggiori quando $\varepsilon = 5\%$, per cui si ipotizza che tutti i ratio siano in grado di trarre profitto dal maggior grado di *exploration*. In particolare Sortino e Burke ottengono rendimenti negativi solo quando utilizzano un basso valore di *exploration*. A proposito, si riportano i grafici per le tre funzioni di reward nel caso di $L = 5$ e $N = 5$ con lo scopo di confrontare le performance quando $\varepsilon = 5\%$ e $\varepsilon = 15\%$. In ordine sono: Sharpe in alto, Burke al centro e Sortino in basso. Il grafico di sinistra è l'output per ogni ratio nel caso di un minor grado di *exploration*, mentre a destra il caso di maggior *exploration*. Osservando i seguenti grafici, si deduce che il maggior valore di ε porta ad un maggior numero di azioni in grado di cogliere i movimenti dei prezzi traendone un profitto. Si può notare come, prevedibilmente, all'aumentare del valore di ε aumentino le operazioni compiute. Mentre con $\varepsilon = 5\%$ si compiono più operazioni nella prima metà del periodo di investimento mentre nella seconda metà molto meno, con $\varepsilon = 15\%$ si compiono in modo più uniforme lungo tutto l'arco del periodo di investimento, soprattutto quando nella seconda metà il

prezzo ha una tendenza positiva. Questo vale in particolare con

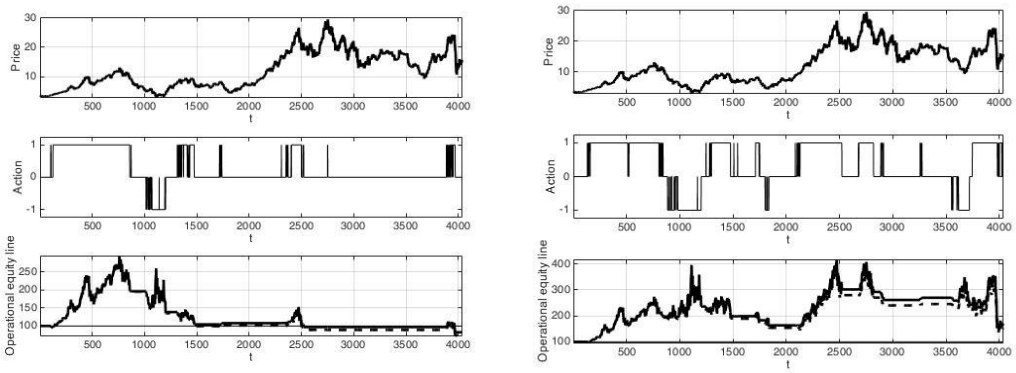

Figura 5.14: Azimut S.p.A., Sharpe ratio QL con N=5, L=5. A sinistra $\varepsilon = 5\%$, a destra $\varepsilon = 15\%$. Fonte: Matlab.

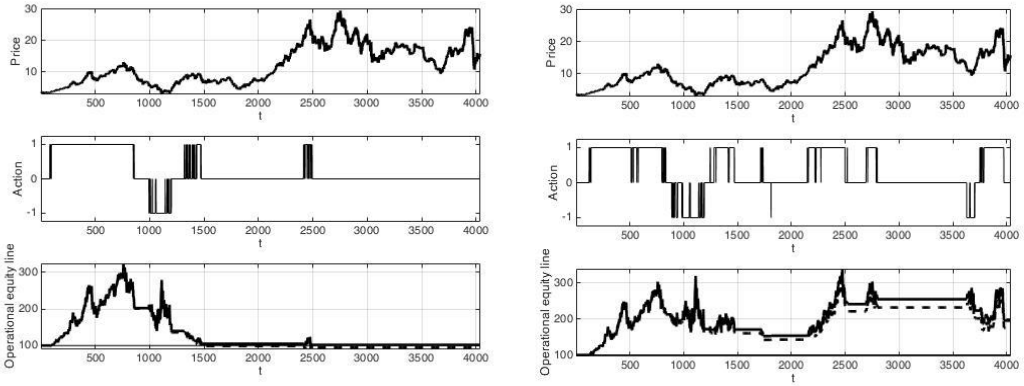

Figura 5.15: Azimut S.p.A., Burke ratio QL con N=5, L=5. A sinistra $\varepsilon = 5\%$, a destra $\varepsilon = 15\%$. Fonte: Matlab.

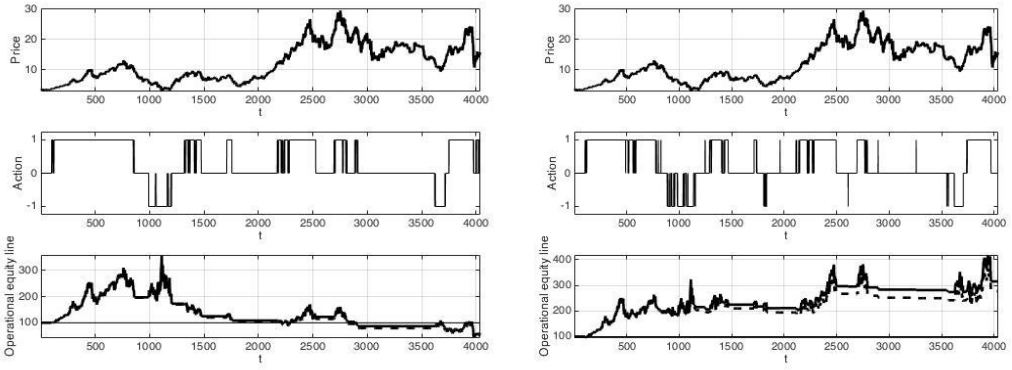

Figura 5.16: Azimut S.p.A., Sortino ratio QL con N=5, L=5. A sinistra $\varepsilon = 5\%$, a destra $\varepsilon = 15\%$. Fonte: Matlab.

Burke ratio, dove nel caso di $\varepsilon = 5\%$ da $t = 1500$ resta fuori dal mercato per la quasi totalità del periodo. Ovviamente, l'aumentare del numero di operazioni non significa che aumentino anche le performance, come avviene nel caso di Burke.

Dai grafici si osserva anche l'andamento dei prezzi del titolo in esame. In modo simile al titolo Amplifon S.p.A. vi è un andamento senza trend nella prima parte, mentre nella seconda vi è un trend rialzista seguito da un periodo di oscillazioni. Questo andamento potrebbe essere la causa della scarsa performance di Sharpe rispetto agli altri ratio. Tuttavia, le *equity lines* si assomigliano tra diversi ratio se confrontati a parità di ε. La differenza di performance dipende dal livello in cui chiudono l'investimento, si noti soprattutto per $\varepsilon = 15\%$ dove tutti e tre terminano con un picco di perdita. Si noti come anche nel caso di Azimut S.p.A. le performance sono più simili tra Sharpe e Sortino, mentre è più accentuata la differenza con Burke.

Le prossime tabelle riportano i risultati relativi ad Azimut S.p.A. ottenuti con l'algoritmo SARSA. In questo caso, Sharpe migliora solo per piccoli ε quando $L = 5$, mentre peggiora decisamente per $L = 22$. Se si guarda alle

N=1	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	114,008	**112,809**	0,003	**0,003**	0,822	**0,755**	99,802	**99,802**	0,9
		10	146,221	**144,478**	0,009	**0,009**	2,400	**2,323**	99,480	**99,430**	1,0
		15	124,327	**121,202**	0,005	**0,005**	1,368	**1,208**	98,266	**98,192**	2,1
	22	5	50,493	**48,487**	-0,017	**-0,018**	-4,176	**-4,418**	71,613	**70,349**	3,4
		10	72,389	**70,239**	-0,008	**-0,009**	-1,997	**-2,181**	84,493	**83,973**	2,5
		15	70,646	**66,229**	-0,009	**-0,010**	-2,146	**-2,539**	45,727	**42,928**	5,4
BR	5	5	141,416	**139,938**	0,009	**0,008**	2,187	**2,120**	99,430	**99,381**	0,9
		10	144,832	**143,319**	0,009	**0,009**	2,339	**2,272**	99,356	**99,282**	0,9
		15	163,031	**159,638**	0,012	**0,012**	3,098	**2,963**	100,000	**99,926**	1,7
	22	5	133,144	**130,190**	0,007	**0,007**	1,803	**1,660**	99,133	**97,399**	1,9
		10	134,482	**132,485**	0,007	**0,007**	1,867	**1,771**	99,752	**99,628**	1,2
		15	136,004	**133,378**	0,008	**0,007**	1,938	**1,814**	95,294	**95,145**	1,6
SOR	5	5	97,218	**92,641**	-0,001	**-0,002**	-0,176	**-0,476**	99,133	**98,811**	4,0
		10	62,903	**56,005**	-0,011	**-0,014**	-2,852	**-3,554**	70,052	**68,863**	9,7
		15	136,928	**122,560**	0,008	**0,005**	1,981	**1,278**	100,000	**100,000**	9,2
	22	5	104,904	**104,436**	0,001	**0,001**	0,299	**0,271**	99,529	**99,505**	0,4
		10	128,102	**126,191**	0,006	**0,006**	1,558	**1,463**	99,405	**99,282**	1,2
		15	135,662	**134,053**	0,008	**0,007**	1,922	**1,846**	99,554	**99,554**	1,0

Tabella 5.7: Azimut S.p.A. per SARSA con N=1.

N=5	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	135,509	**135,307**	0,008	**0,007**	1,917	**1,907**	100,000	**100,000**	0,1
		10	119,093	**115,741**	0,004	**0,004**	1,098	**0,918**	99,380	**99,355**	2,4
		15	127,266	**123,682**	0,006	**0,005**	1,518	**1,337**	99,033	**98,810**	2,4
	22	5	89,052	**86,934**	-0,003	**-0,003**	-0,722	**-0,871**	88,768	**88,594**	2,0
		10	101,761	**97,278**	0,000	**-0,001**	0,109	**-0,172**	99,678	**87,057**	3,7
		15	118,011	**106,724**	0,004	**0,002**	1,040	**0,407**	100,000	**99,901**	8,4
BR	5	5	151,170	**149,815**	0,010	**0,010**	2,616	**2,558**	99,826	**99,826**	0,7
		10	152,655	**151,284**	0,010	**0,010**	2,678	**2,621**	100,000	**99,975**	0,7
		15	179,738	**178,129**	0,015	**0,014**	3,732	**3,673**	99,926	**99,926**	0,7
	22	5	167,014	**164,280**	0,013	**0,012**	3,257	**3,150**	99,033	**97,992**	1,4
		10	172,118	**169,802**	0,013	**0,013**	3,451	**3,364**	99,157	**99,083**	1,1
		15	181,415	**178,976**	0,015	**0,014**	3,792	**3,704**	99,405	**99,405**	1,1
SOR	5	5	122,777	**116,132**	0,005	**0,004**	1,290	**0,939**	99,430	**99,405**	4,6
		10	92,203	**81,149**	-0,002	**-0,005**	-0,506	**-1,297**	97,917	**80,883**	10,6
		15	78,731	**70,657**	-0,006	**-0,009**	-1,483	**-2,147**	72,204	**71,535**	9,0
	22	5	120,590	**119,329**	0,005	**0,004**	1,177	**1,110**	100,000	**100,000**	0,9
		10	149,922	**148,579**	0,010	**0,010**	2,563	**2,505**	99,926	**99,926**	0,7
		15	155,076	**152,526**	0,011	**0,010**	2,779	**2,673**	100,000	**100,000**	1,4

Tabella 5.8: Azimut S.p.A. per SARSA con N=5.

percentuali over 100, si nota che con $L = 22$ e $\varepsilon = 15\%$ è pari a 42,928% mentre negli altri casi è comunque superiore al 70%. Questo attenua un po' la performance negativa che può dirsi per lo più con $\varepsilon = 15\%$. Con Burke ratio le performance sono positive in tutte le combinazioni di parametri con percentuali over 100 prossime al 100%. I rendimenti sono poi maggiori in $L = 5$ rispetto a $L = 22$, che potrebbe indicare che Burke ratio con SARSA sia in grado di reagire tempestivamente alle variazioni di rendimento, dato che con meno rendimenti considerati ottiene risultati maggiori. Si noti però che il numero di operazioni è più bassi rispetto al QL in particolare per Burke. Il caso di Sortino sembra più simile a Burke piuttosto che a Sharpe, dato che dei due valori negativi di rendimento, uno ha la percentuale over 100 pari al 99% per cui l'investimento è terminato in un momento in cui il prezzo è sceso al 100, ma in generale la performance è stata positiva. L'unico valore di perdita sostanziosa si ha per $L = 5$ e $\varepsilon = 10\%$. Quindi le performance migliori in termini di rendimenti netti si sono ottenute con Burke, con anche il valore in assoluto più alto della tabella per un capitale finale netto di 159,638€ pari ad un rendimento annuo netto di 2,963%.

Nel caso di Sharpe $L = 5$ i risultati sono in linea con il caso $N = 1$ ma migliorano di valore sia in termini di rendimento sia di percentuali. Anche se rimangono negativi per $L = 22$ la perdita è decisamente minore.

Guardando alcuni grafici è possibile approfondire il comportamento degli indici nel caso di SARSA. Si veda la seguente Figura 5.17. Sono state selezionate due combinazioni di Burke il più possibile diverse tra loro: a sinistra il risultato con $N = 1$, $L = 5$, $\varepsilon = 5\%$ mentre a destra $N = 5$, $L = 22$, $\varepsilon = 15\%$. L'unica differenza che si può notare è che nel grafico di destra l'algoritmo compie qualche azione in più rispetto al grafico di sinistra, presumibilmente dovute al fatto che ε è maggiore. In generale però le due *equity lines* e le azioni che sono state intraprese sono molto simili. Quindi Burke sicuramente ha ottenuto i risultati migliori in termini di rendimenti e di percentuali over 100, tuttavia li ha ottenuti stando fuori

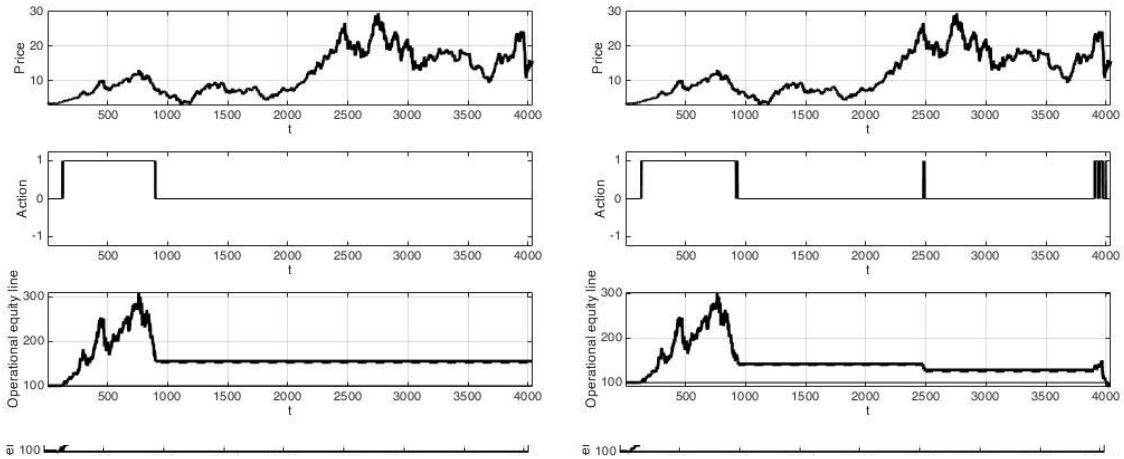

Figura 5.18: Azimut S.p.A., Sortino ratio SARSA. A sinistra: N=1, L=5, $\varepsilon = 5\%$. A destra: N=5, L=22, $\varepsilon = 5\%$. Fonte: Matlab.

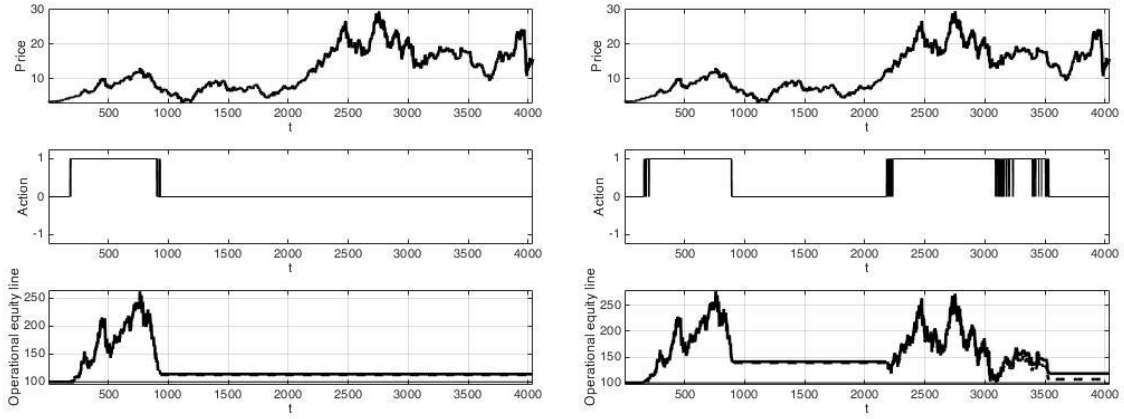

Figura 5.19: Azimut S.p.A., Sharpe ratio SARSA. A sinistra: N=1, L=5, $\varepsilon = 5\%$. A destra: N=5, L=22, $\varepsilon = 5\%$. Fonte: Matlab

dal mercato per la maggior parte del tempo mostrando quindi una certa rigidità, più accentuata in SARSA che in QL. Risultati simili si sono realizzati anche con Sortino e Sharpe illustrati rispettivamente nelle Figure 5.18 e 5.19.
Si può concludere che nel caso del titolo Azimut S.p.A. con l'algoritmo QL si sono realizzate performance in generale positive, con ottimi valori di percentuali over 100 per tutti i ratio e gli algoritmi. Tra $N = 1$ ed $N = 5$ si realizzano ni generale dei miglioramenti con $N = 5$ in maggior misura per il Burke ratio. È possibile affermare anche che le funzioni di reward ottengono le performance peggiori quando $\varepsilon = 5\%$, per cui è possibile che siano in grado di trarre maggior profitto quando aumenta il grado di *exploration*. Per quanto riguarda i risultati con SARSA, ci sono miglioramenti di performance quando $N = 5$, accompagnati anche da ottimi valori di percentuali over 100 che in cinque casi sono pari al 100%. Se si confrontano i valori tra QL e SARSA non c'è un algoritmo che performa meglio dell'altro in modo evidente. È possibile affermare soltanto che SARSA mostra rigidità nel compiere le azioni, dato che in 17 combinazioni su 18 totali (sia per $N = 1$ che per $N = 5$) QL ha un numero di operazioni annuo maggiore rispetto a SARSA. Per esempio, con l'algoritmo QL $N = 1$ la media delle azioni considerando tutte le funzioni di reward è circa di 9 operazioni l'anno, nel caso di SARSA la media è di circa 2 operazioni. Lo stesso risulta con $N = 5$. Confrontando solo i rendimenti invece, risulta che per QL in $N = 1$ gli indici di Burke e Sortino ottengano valori migliori di Sharpe, mentre in $N = 5$ Burke risulta migliore in generale anche di Sortino. Lo stesso vale SARSA dove in particolare per $N = 5$ Burke ottiene i rendimenti migliori in assoluto.

5.1.3 Banco BPM S.p.A.

Nella prima tabella si presentano i risultati per QL con $N = 1$ del titolo Banco BPM. Nel caso di Sharpe si osserva che per $L = 5$ c'è un rendimento negativo a cui però si associa un percentuale over 100 piuttosto alta. Per $L = 22$ invece ottiene tre rendimenti negativi, in particolare con $\varepsilon = 15\%$ dove il rendimento annuo netto è di $-5{,}549\%$ che corrisponde ad un capitale finale netto di 40,069€ ed è la performance peggiore della tabella. Nel caso di Burke invece si ottengono

risultati migliori poichè le percentuali sono prossime al 100% e i rendimenti sono positivi, in particolare per $L = 5$. Un rendimento negativo si realizza con $L = 22$ $\varepsilon = 15\%$ di circa il -4%. Sortino invece presenta rendimenti positivi e negativi, tuttavia migliori per $L = 22$ se paragonati a $L = 5$ (a parità di ε). Anche in questo caso il rendimento più basso si ottiene per $L = 22$ e $\varepsilon = 15\%$ come per Sharpe e Burke.

N=1	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	196,801	**190,961**	0,017	**0,016**	4,317	**4,121**	99,009	**98,885**	2,5
		10	69,695	**63,219**	-0,009	**-0,011**	-2,229	**-2,822**	81,521	**80,778**	8,1
		15	145,784	**132,071**	0,009	**0,007**	2,381	**1,752**	99,876	**99,851**	8,2
	22	5	63,237	**57,634**	-0,011	**-0,014**	-2,820	**-3,381**	73,743	**72,356**	7,7
		10	63,946	**57,313**	-0,011	**-0,014**	-2,753	**-3,415**	79,589	**77,434**	9,1
		15	47,321	**40,069**	-0,019	**-0,023**	-4,563	**-5,549**	61,927	**61,580**	13,9
BR	5	5	158,343	**150,721**	0,011	**0,010**	2,910	**2,594**	99,653	**97,746**	4,1
		10	136,872	**124,680**	0,008	**0,005**	1,979	**1,386**	95,219	**94,526**	7,7
		15	165,742	**147,456**	0,013	**0,010**	3,204	**2,454**	99,678	**99,529**	9,7
	22	5	117,340	**108,677**	0,004	**0,002**	1,003	**0,521**	94,327	**93,882**	6,4
		10	118,003	**109,466**	0,004	**0,002**	1,039	**0,566**	94,996	**94,922**	6,2
		15	59,533	**52,009**	-0,013	**-0,016**	-3,186	**-3,999**	61,258	**54,595**	11,2
SOR	5	5	96,543	**91,332**	-0,001	**-0,002**	-0,219	**-0,564**	79,812	**79,688**	4,6
		10	106,906	**97,214**	0,002	**-0,001**	0,418	**-0,176**	99,579	**78,251**	7,9
		15	231,137	**204,413**	0,021	**0,018**	5,369	**4,564**	97,622	**94,996**	10,2
	22	5	103,287	**93,408**	0,001	**-0,002**	0,202	**-0,425**	99,876	**77,434**	8,4
		10	151,101	**135,884**	0,010	**0,008**	2,610	**1,932**	99,975	**99,975**	8,9
		15	76,268	**66,934**	-0,007	**-0,010**	-1,677	**-2,475**	23,730	**23,483**	10,9

Tabella 5.9: Banco BPM S.p.A. per QL con N=1

Dalla Tabella 5.10 si osservano i risultati per $N = 5$. Con Sharpe si ottiene un miglioramento per $L = 22$ e $\varepsilon \in \{5\%, 10\%\}$ che da rendimenti negativi ora diventano positivi, invece con $\varepsilon = 15\%$ resta negativo ma passa da un valore di $-5{,}549\%$ al valore di $-1{,}105\%$, cioè da un capitale finale netto di 47,321€ a 83,707%. Burke ratio invece per $\varepsilon = 5\%$ per entrambi i valori di L migliora le performance già positive, sia in termini di rendimenti sia di percentuali over 100%. Negli altri casi invece peggiora i suoi risultati che mentre con $L = 5$ restano positivi con alti valori di percentuale over 100, invece con $L = 22$ diventano negativi o ancora più negativi, dove le percentuali over 100 peggiorano. In generale risulta da questi dati che Sortino sia l'indice con

performance maggiori a parità di parametri, soprattutto con $N = 5$ dove Sortino ottiene dei rendimenti annui netti maggiori in 4 casi su 6 e gli altri due invece sono attribuiti a Burke. A prima vista sembra quindi che nel caso del titolo Banco BPM le performance di Sharpe siano le peggiori.

N=5	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	186,625	174,674	0,015	0,014	3,976	**3,547**	97,471	96,281	5,5
		10	92,794	83,150	-0,002	-0,005	-0,466	**-1,146**	78,651	78,378	9,1
		15	136,690	126,401	0,008	0,006	1,972	**1,475**	99,603	99,529	6,5
	22	5	145,474	138,467	0,009	0,008	2,370	**2,054**	99,355	99,083	4,1
		10	116,017	109,418	0,004	0,002	0,933	**0,564**	100,000	100,000	4,9
		15	93,262	83,707	-0,002	-0,004	-0,435	**-1,105**	76,965	58,220	9,0
BR	5	5	245,615	237,675	0,022	0,021	5,775	**5,558**	100,000	100,000	2,7
		10	135,496	124,165	0,008	0,005	1,916	**1,362**	94,396	94,396	7,2
		15	134,688	121,259	0,007	0,005	1,878	**1,212**	97,967	97,396	8,7
	22	5	162,754	152,379	0,012	0,010	3,090	**2,667**	94,148	93,776	5,5
		10	72,454	63,269	-0,008	-0,011	-1,993	**-2,820**	55,914	44,409	11,2
		15	56,858	49,685	-0,014	-0,017	-3,466	**-4,276**	26,630	24,622	11,2
SOR	5	5	117,426	111,586	0,004	0,003	1,009	**0,687**	100,000	99,851	4,2
		10	147,715	137,408	0,010	0,008	2,468	**2,005**	99,653	99,653	6,0
		15	162,382	146,846	0,012	0,010	3,075	**2,430**	97,074	95,115	8,4
	22	5	92,129	83,578	-0,002	-0,004	-0,511	**-1,115**	68,287	43,342	8,1
		10	122,786	116,336	0,005	0,004	1,291	**0,950**	99,926	99,851	4,5
		15	89,019	84,336	-0,003	-0,004	-0,724	**-1,059**	29,060	28,267	4,5

Tabella 5.10: Banco BPM S.p.A. per QL con N=5

Le Tabelle 5.11 e 5.12 mostrano i risultati per l'algoritmo SARSA. Si nota facilmente che tutti i rendimenti risultano negativi. Confrontando con i risultati appena analizzati di QL, si nota che le performance rimangono migliori con QL che ottiene 13 valori maggiori su 18 combinazioni totali. Tuttavia in alcuni casi negativi di QL, SARSA ottiene risultati migliori a cui vengono tuttavia associate delle percentuali over 100 piuttosto basse: mentre quelle nel caso di Q- *Learning* sono in media maggiori dell'80%, nel caso di SARSA sono in media del 26%. Questo indica che, anche se a livello di rendimenti SARSA ottiene valori più alti, non significa che siano frutto dell'abilità di SARSA. In più, si nota che Sharpe non ottiene performance migliori delle altre misure in nessuna delle combinazioni presentate, mentre Burke ottiene le performance migliori quando $N = 1$ e Sortino quando $N = 5$. I rendimenti sono comunque bassi in SARSA dove il valore

massimo che si ottiene è pari ad un capitale finale netto di 114,233€ pari ad un rendimento annuo netto di 0,834% nel caso di Sortino

N=1	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	64,924	56,127	-0,011	-0,014	-2,660	**-3,541**	19,123	19,074	12,1
		10	36,453	31,105	-0,025	-0,029	-6,105	**-7,030**	19,321	18,826	13,2
		15	74,442	63,614	-0,007	-0,011	-1,826	**-2,784**	24,028	21,476	13,1
	22	5	39,658	37,579	-0,023	-0,024	-5,610	**-5,927**	22,046	21,625	4,5
		10	64,898	63,647	-0,011	-0,011	-2,663	**-2,781**	33,713	33,614	1,6
		15	50,513	46,573	-0,017	-0,019	-4,174	**-4,658**	25,985	24,498	6,7
BR	5	5	94,681	92,706	-0,001	-0,002	-0,341	**-0,472**	45,009	44,934	1,7
		10	101,382	95,617	0,000	-0,001	0,086	**-0,279**	96,631	14,491	4,9
		15	99,900	90,199	0,000	-0,003	-0,006	**-0,642**	47,709	21,055	8,5
	22	5	91,499	86,949	-0,002	-0,003	-0,553	**-0,869**	14,119	13,228	4,2
		10	91,698	88,584	-0,002	-0,003	-0,540	**-0,754**	15,135	14,912	2,9
		15	65,989	57,372	-0,010	-0,014	-2,561	**-3,409**	47,511	47,238	11,6
SOR	5	5	47,052	42,280	-0,019	-0,021	-4,597	**-5,232**	29,254	28,065	8,9
		10	123,346	114,233	0,005	0,003	1,318	**0,834**	76,839	75,006	6,4
		15	52,828	47,195	-0,016	-0,019	-3,905	**-4,579**	19,519	17,488	9,4
	22	5	76,371	76,023	-0,007	-0,007	-1,669	**-1,697**	19,891	19,891	0,4
		10	88,926	88,260	-0,003	-0,003	-0,730	**-0,777**	19,767	19,693	0,6
		15	80,402	79,323	-0,005	-0,006	-1,352	**-1,436**	18,058	17,191	1,1

Tabella 5.11: Banco BPM S.p.A. per SARSA con N=1

$L = 5$ e $\varepsilon = 10\%$. Maggiori in generale con QL in Tabella 5.9 che raggiunge rendimenti spesso sopra al massimo ottenuto con SARSA, in particolare il maggior valore si raggiunge con Sortino $L = 5$ però con $\varepsilon = 15\%$ per un capitale finale netto di 204,413€ pari a un rendimento di 4,564%.

Più accentuata la differenza di rendimenti tra gli algoritmi se si osserva la Tabella 5.12 che raccoglie i risultati di SARSA con $N = 5$. Confrontandola con la Tabella 5.10 per QL $N = 5$ si nota ancora più del caso $N = 1$ che le performance sono migliori per QL. Infatti, SARSA ottiene rendimenti migliori anche in questo caso per ε maggiori, ma solo per Burke e Sortino quando $L = 22$. Sharpe invece in tutti i casi $N = 5$ ottiene valori migliori con QL, in particolare si noti il caso di $L = 22$ e $\varepsilon = 5\%$ dove Sharpe raggiunge un rendimento annuo netto di $-13{,}712\%$ pari ad un capitale finale netto di 9,440€ su un investimento iniziale di 100€. Invece se si confrontano i risultati di SARSA tra loro, cioè le Tabelle 5.11 e 5.12, non sembra esserci un caso migliore dell'altro in modo evidente.

N=5	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	L	ε (%)	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	#
SR	5	5	87,064	86,671	-0,003	-0,004	-0,862	**-0,890**	19,266	19,216	0,4
		10	79,418	74,006	-0,006	-0,007	-1,430	**-1,863**	18,373	18,175	5,9
		15	45,071	39,437	-0,020	-0,023	-4,858	**-5,648**	15,745	15,522	11,1
	22	5	11,987	9,440	-0,053	-0,059	-12,414	**-13,712**	19,415	19,390	19,9
		10	35,460	26,510	-0,026	-0,033	-6,273	**-7,961**	19,762	19,117	24,2
		15	28,660	24,513	-0,031	-0,035	-7,511	**-8,410**	19,117	19,092	13,0
BR	5	5	94,876	92,488	-0,001	-0,002	-0,328	**-0,487**	18,423	17,679	2,1
		10	102,276	101,205	0,001	0,000	0,141	**0,075**	99,826	99,777	0,9
		15	65,496	59,137	-0,010	-0,013	-2,610	**-3,229**	12,447	12,150	8,5
	22	5	60,675	57,215	-0,012	-0,014	-3,074	**-3,429**	14,109	13,836	4,9
		10	94,047	93,066	-0,002	-0,002	-0,383	**-0,448**	14,233	14,084	0,9
		15	116,997	101,589	0,004	0,000	0,986	**0,099**	70,121	68,733	11,7
SOR	5	5	113,735	111,542	0,003	0,003	0,807	**0,685**	76,370	76,147	1,6
		10	107,404	102,380	0,002	0,001	0,447	**0,147**	86,065	85,867	4,0
		15	96,904	89,098	-0,001	-0,003	-0,196	**-0,719**	22,316	20,035	7,0
	22	5	73,302	71,987	-0,008	-0,008	-1,922	**-2,033**	19,812	19,812	1,5
		10	86,191	84,906	-0,004	-0,004	-0,924	**-1,017**	19,663	19,638	1,2
		15	85,347	84,836	-0,004	-0,004	-0,985	**-1,022**	19,068	18,795	0,5

Tabella 5.12: Banco BPM S.p.A. per SARSA con N=5

Per concludere, sia in QL che in SARSA non ci sono differenze di performance al variare del parametro N. Le differenze sono più evidenti tra performance attribuite all'algoritmo, dove QL sembra più performante di SARSA sia in termini di rendimenti, sia di percentuali over 100. Tra le misure di performance, la migliore in termini di rendimenti annui netti è Sortino, seguito da Burke. A differenza degli altri titoli, questo è il primo caso in cui Sharpe risulta il peggiore tra i ratio. Si prova a confrontare dunque dei grafici a tre pannelli con una combinazione di parametri, per esempio $L = 5, N = 1, \varepsilon = 10\%$, illustrati nelle figure $5.20, 5.21, 5.22$. Si osservano diverse differenze tra i due algoritmi in generale per tutte le misure di performance. Concentrandosi solo su Sharpe ratio, nonostante gli input ai due algoritmi siano gli stessi, si nota come agiscono diversamente. SARSA concentra diverse azioni di vendita quando il prezzo lievemente scende, accumulando una perdita che mantiene poi per il resto del tempo di investimento. Nel caso di QL compie azioni di vendita e di acquisto lungo tutto il periodo di investimento e con una concentrazione minore rispetto a

SARSA, gestendo meglio il periodo di discesa del prezzo che in SARSA genera una perdita importante. Il caso di Burke richiama quanto visto in precedenza con gli altri titoli: il numero di operazioni annue compiuto è minore delle altre funzioni di reward e si accentua quando si utilizza l'algoritmo SARSA. Dal grafico di destra infatti si vedono alcune operazioni di acquisto nel periodo circa $t\epsilon[500{,}800]$ prima e dopo il quale resta costantemente fuori dal mercato. A differenza di Burke invece, Sortino assomiglia all'indice di Sharpe soprattutto con l'algoritmo QL. La differenza di performance tra i due in SARSA è dovuta al fatto che Sortino riesce a cogliere nel modo corretto i segnali dove Sharpe invece genera delle perdite che mantiene fino al termine. Questo potrebbe essere dovuto al fatto che la linea dei oscilla senza trend da metà periodo in poi, cioè dal punto in cui Sharpe accumula perdite e dove Sortino invece crea guadagni. La differenza consiste nella natura della volatilità, come già visto per Amplifon S.p.A. che comporta un modo diverso di reagire alle oscillazioni. Sharpe considera sia scostamenti negativi che positivi, mentre Sortino considera solo le perdite. Con QL i risultati tra Sharpe e Sortino sono molto simili (come già visto in altri titoli) mentre utilizzando SARSA sono molto diversi. Infine, anche in questo titolo Burke mostra una rigidità che risulta più evidente quando si utilizza l'algoritmo SARSA. Si noti inoltre la differenza di performance finale di Burke con SARSA causata dai costi di transazione: senza costi l'investimento termina positivamente, considerando i costi invece l'investimento realizza un risultato negativo.

5.1.4 Campari S.p.A.

Dalla Tabella 5.13, contenente i risultati per QL per $N = 1$, risultano in generale migliori i rendimenti con Sharpe, soprattutto quando $\varepsilon \in \{10\%, 15\%\}$. Se si osserva il valore per $L = 22$ e $\varepsilon = 15\%$ di tutti e tre le misure, si nota come nessuna di queste sia stata in grado di gestire il maggior valore di *exploration* quando si tengono in considerazione più rendimenti passati ($L = 22$). Inoltre per questi parametri le performance sono accompagnate da percentuali over 100 molto basse, che vuol dire che non solo è terminata con rendimenti negativi per tutti, ma in generale l'investimento è stato negativo nella maggior parte dell'intero

periodo. Si noti come Burke e Sortino in particolare non riescano ad ottenere rendimenti

N=1	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	***N €***	*G %*	***N %***	*G %*	***N %***	*G %*	***N %***	*#*
SR	5	5	107,054	102,178	0,002	0,001	0,426	0,135	99,108	99,108	3,9
		10	151,219	137,987	0,010	0,008	2,615	2,030	77,087	50,012	7,6
		15	124,265	109,706	0,005	0,002	1,365	0,580	46,074	41,169	10,4
	22	5	161,913	150,213	0,012	0,010	3,054	2,572	53,183	44,934	6,2
		10	202,252	182,110	0,017	0,015	4,495	3,813	94,154	89,002	8,7
		15	91,479	75,954	-0,002	-0,007	-0,554	-1,702	9,388	7,629	15,5
BR	5	5	110,489	107,861	0,002	0,002	0,625	0,474	97,102	97,003	2,0
		10	131,580	117,386	0,007	0,004	1,728	1,006	47,857	41,813	9,5
		15	74,222	60,605	-0,007	-0,012	-1,844	-3,078	30,295	28,437	16,9
	22	5	204,506	184,694	0,018	0,015	4,567	3,904	99,827	99,827	8,5
		10	107,559	89,186	0,002	-0,003	0,456	-0,712	86,673	40,921	15,6
		15	66,149	53,053	-0,010	-0,016	-2,547	-3,880	27,471	19,941	18,4
SOR	5	5	115,931	108,679	0,004	0,002	0,927	0,521	99,827	99,827	5,4
		10	129,486	114,665	0,006	0,003	1,626	0,858	66,237	57,394	10,1
		15	57,705	46,903	-0,014	-0,019	-3,374	-4,616	26,975	25,910	17,2
	22	5	49,945	38,932	-0,017	-0,023	-4,241	-5,719	30,245	26,827	20,7
		10	87,722	75,597	-0,003	-0,007	-0,814	-1,731	42,135	36,413	12,4
		15	86,982	75,088	-0,003	-0,007	-0,867	-1,773	38,865	9,289	12,2

Tabella 5.13: Campari S.p.A. per QL con N=1

positivi con $\varepsilon = 15\%$ per ogni L. Dalla seconda tabella si osservano i risultati per QL con $N = 5$. Da un primo confronto di rendimenti annui netti si evince che in 12 valori su 18 totali $N = 5$ abbia performance migliori. Questo vale per Sharpe e in particolare per Sortino, che migliora tutti e sei i suoi valori. Risulta in questa tabella quindi che le performance migliori siano attribuite a Sharpe ed a Sortino. Tuttavia, il valore di rendimento maggiore si ottiene in $N = 1$ con Sharpe quando $L = 22$ ed $\varepsilon = 10\%$ con un rendimento di 3,813% pari ad un capitale di 182,110€. Questi risultati suggeriscono dunque che sia Sortino che Sharpe lavorano meglio quando la struttura degli stati racchiude informazioni su più giorni di trading passati piuttosto che sul loro giorno prima. Questo non vale invece con Burke che sembra performa meglio con $N = 5$ sono per $L = 22$. Si noti però che il maggior valore di *N* non diventa un vantaggio nel caso di maggior *exploration* per Sharpe e Burke.

N=5	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	136,425	134,397	0,008	0,007	1,960	1,864	99,851	99,826	1,2
		10	166,354	154,104	0,013	0,011	3,231	2,739	51,897	47,384	6,4
		15	140,021	122,726	0,008	0,005	2,126	1,288	35,457	29,308	11,0
	22	5	178,921	163,496	0,014	0,012	3,702	3,120	87,602	81,031	7,5
		10	123,091	105,473	0,005	0,001	1,307	0,333	60,030	48,078	12,9
		15	48,362	37,968	-0,018	-0,024	-4,438	-5,872	8,406	7,439	20,1
BR	5	5	133,068	131,675	0,007	0,007	1,801	1,734	98,066	97,942	0,9
		10	137,330	122,317	0,008	0,005	2,002	1,267	41,334	33,424	9,6
		15	55,016	44,991	-0,015	-0,020	-3,665	-4,868	25,886	24,473	16,7
	22	5	107,319	93,606	0,002	-0,002	0,442	-0,412	79,866	51,426	11,4
		10	83,105	66,541	-0,005	-0,010	-1,150	-2,513	26,333	24,944	18,5
		15	46,361	37,064	-0,019	-0,025	-4,690	-6,013	23,456	17,158	18,6
SOR	5	5	143,425	137,108	0,009	0,008	2,279	1,992	99,876	99,876	3,7
		10	151,278	136,163	0,010	0,008	2,620	1,948	79,916	77,015	8,7
		15	62,057	49,247	-0,012	-0,018	-2,937	-4,329	27,349	25,886	19,2
	22	5	91,662	74,855	-0,002	-0,007	-0,543	-1,793	33,325	30,126	16,9
		10	114,824	109,113	0,003	0,002	0,867	0,546	98,983	98,934	4,2
		15	83,344	78,723	-0,005	-0,006	-1,132	-1,484	6,695	6,174	4,7

Tabella 5.14: Campari S.p.A. per QL con N=5

Dalle Tabelle 5.15 e 5.16 si osservano i risultati relativi al titolo Campari S.p.A. con SARSA, rispettivamente per $N = 1$ e $N = 5$. Da un primo confronto con le tabelle precedenti relative a QL risulta che SARSA ottiene rendimenti migliori di QL, soprattutto nella Tabella 5.15 (cioè SARSA $N = 1$) dove migliorano tutte le combinazioni ad eccezione di una (Burke $L = 22$ $\varepsilon = 5\%$). Anche in questi due casi a giocarsi i rendimenti più alti sono Sharpe e Sortino, nonostante anche Burke ottenga rendimenti positivi con ottime percentuali over 100. Più evidenti dei rendimenti sono i confronti tra le percentuali over 100: con SARSA infatti le percentuali migliorano nella quasi totalità dei casi avvicinandosi a valori prossimi al 100%, in modo molto simile tra $N = 1$ ed $N = 5$. Confrontando tra loro le tabelle con i risultati ottenuti con SARSA non sembra ci siano sostanziali differenza tra i valori di N.

N=1	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	145,779	143,830	0,009	0,009	2,381	2,295	99,480	99,331	1,1
		10	144,779	143,264	0,009	0,009	2,337	2,270	99,331	99,158	0,9
		15	135,968	134,752	0,008	0,007	1,936	1,879	99,653	98,836	0,7
	22	5	213,492	205,155	0,019	0,018	4,848	4,588	70,523	65,147	3,3
		10	241,714	220,433	0,022	0,020	5,664	5,058	97,498	97,250	7,7
		15	124,802	104,164	0,005	0,001	1,393	0,255	27,719	20,188	15,0
BR	5	5	145,199	144,109	0,009	0,009	2,355	2,307	98,365	98,142	0,6
		10	148,029	147,368	0,010	0,010	2,479	2,450	99,579	99,554	0,4
		15	144,260	143,186	0,009	0,009	2,314	2,266	99,554	99,381	0,6
	22	5	115,879	104,342	0,004	0,001	0,924	0,266	99,356	93,882	8,7
		10	113,391	111,704	0,003	0,003	0,788	0,693	100,000	100,000	1,2
		15	120,909	117,692	0,005	0,004	1,192	1,022	100,000	100,000	2,2
SOR	5	5	158,447	158,211	0,011	0,011	2,915	2,905	99,827	99,827	0,1
		10	156,987	156,754	0,011	0,011	2,855	2,846	99,777	99,777	0,1
		15	143,769	141,844	0,009	0,009	2,292	2,206	99,183	99,083	1,1
	22	5	93,270	83,456	-0,002	-0,004	-0,434	-1,123	23,359	17,909	9,2
		10	108,354	106,738	0,002	0,002	0,502	0,408	97,919	97,894	1,2
		15	85,934	83,263	-0,004	-0,005	-0,942	-1,137	10,528	9,289	2,6

Tabella 5.15: Campari S.p.A. per SARSA con N=1

N=5	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	130,821	129,065	0,007	0,006	1,693	1,607	99,331	99,306	1,1
		10	143,364	140,597	0,009	0,008	2,276	2,152	99,182	98,959	1,6
		15	104,693	98,600	0,001	0,000	0,287	-0,088	100,000	14,257	5,0
	22	5	298,556	293,881	0,027	0,027	7,073	6,968	99,702	99,678	1,3
		10	236,013	224,761	0,021	0,020	5,512	5,191	76,891	73,271	4,1
		15	127,022	107,122	0,006	0,002	1,506	0,431	27,573	22,911	14,2
BR	5	5	146,549	145,233	0,009	0,009	2,417	2,359	98,463	98,364	0,7
		10	140,382	138,708	0,008	0,008	2,142	2,066	97,868	97,644	1,0
		15	134,363	131,570	0,007	0,007	1,863	1,729	98,091	97,917	1,7
	22	5	121,411	119,785	0,005	0,004	1,220	1,134	99,430	99,355	1,1
		10	116,736	115,170	0,004	0,004	0,972	0,886	98,760	98,463	1,1
		15	123,119	110,674	0,005	0,003	1,308	0,636	99,826	99,777	8,9
SOR	5	5	159,935	159,697	0,012	0,012	2,978	2,968	99,876	99,876	0,1
		10	158,447	158,211	0,011	0,011	2,918	2,908	99,826	99,826	0,1
		15	138,815	136,742	0,008	0,008	2,070	1,975	98,463	98,314	1,2
	22	5	93,663	84,449	-0,002	-0,004	-0,408	-1,051	94,768	38,706	8,6
		10	107,605	105,677	0,002	0,001	0,459	0,346	99,554	99,355	1,5
		15	102,621	95,065	0,001	-0,001	0,162	-0,316	96,281	12,695	6,4

Tabella 5.16: Campari S.p.A. per SARSA con N=5

Si osservino le tre figure (5.23, 5.24, 5.25) che riportano i grafici a tre pannelli per $L = 22, N = 1, \varepsilon = 10\%$. A sinistra l'output per QL, a destra per SARSA. Le azioni intraprese tra i tre grafici QL si assomigliano, con la differenza che Sharpe mantiene alcune posizioni per più tempo, mentre Burke e Sortino le chiudono prima. Dai grafici relativi a SARSA invece il numero di azioni è notevolmente più basso, tuttavia Sharpe entra nel mercato in posizione di acquisto e ci rimane la quasi totalità della seconda metà del periodo di investimento, ottenendo rendimenti positivi visto l'andamento crescente del prezzo del titolo Campari S.p.A., mentre Burke e Sortino entrano in posizione di acquisto nel primo periodo per circa un intervallo di tempo di 1000 giorni di trading, poi rimangono fuori dal mercato per il resto del tempo di investimento. Questo è il motivo per il quale SARSA performa meglio nella maggior parte dei casi. Come si vede dal confronto dei grafici QL e SARSA, in particolare per Burke e Sortino, il rendimento positivo che ottengono nel primo periodo, uguale nei due algoritmi, viene mantenuto nel caso di SARSA mentre viene perso poco dopo nel caso di QL a causa di alcune posizioni di vendita. Si ripete anche in questo titolo quindi una certa rigidità nella scelta delle azioni quando di utilizza SARSA. Ad influire alle perdite inoltre ci sono i costi di transazione, soprattutto per Burke, dove con QL i rendimenti lordi sono positivi, quelli netti sono negativi.

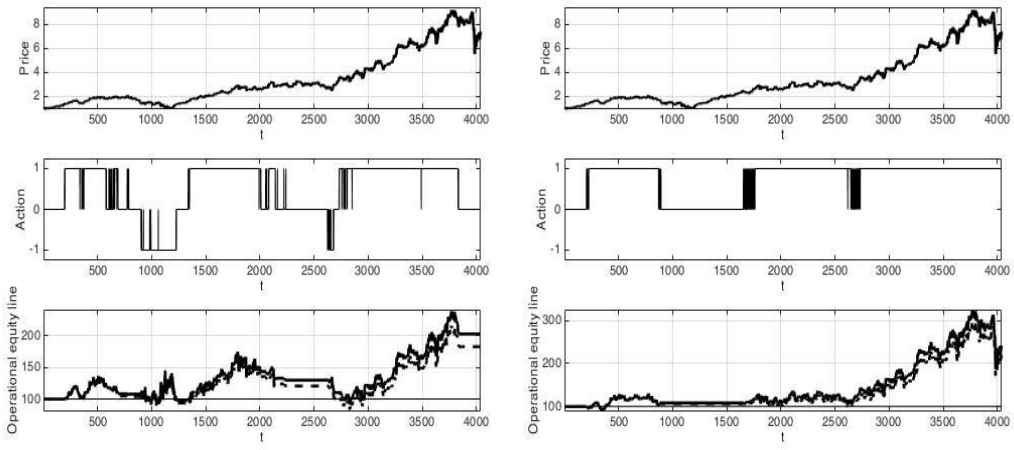

Figura 5.23: Campari S.p.A., Sharpe ratio, N=1, L=22, ε = 10%. A sinistra: QL. A destra: SARSA. Fonte: Matlab.

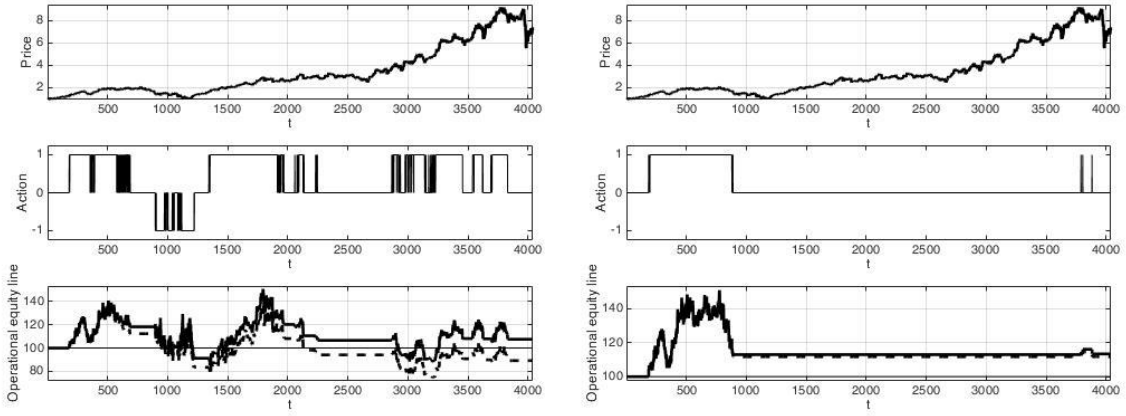

Figura 5.24: Campari S.p.A., Burke ratio, N=1, L=22, $\varepsilon = 10\%$. A sinistra: QL. A destra: SARSA. Fonte: Matlab.

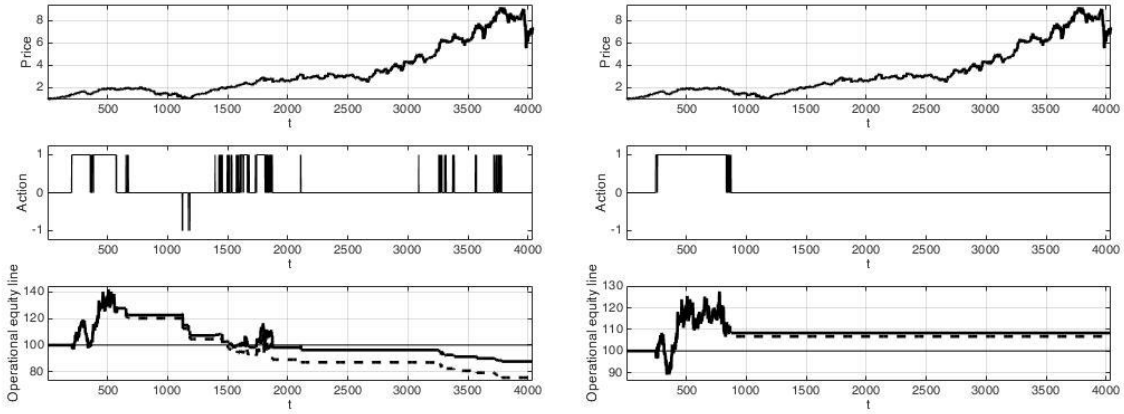

Figura 5.25: Campari S.p.A., Sortino ratio, N=1, L=22, $\varepsilon = 10\%$. A sinistra: QL. A destra: SARSA. Fonte: Matlab

5.1.5 HERA S.p.A.

Dalla prima Tabella 5.17 si nota che Sharpe performa meglio con $L = 5$ piuttosto che $L = 22$, dove con il primo ottiene rendimenti positivi tranne per $\varepsilon = 5\%$, mentre con il secondo ottiene solo rendimenti negativi. Si noti che le percentuali over 100 sono più basse in entrambi i valori di L quando $\varepsilon = 5\%$. Potrebbe quindi indicare, ancora una volta, che Sharpe sia in grado di performare meglio con maggiori valori di *exploration*. Anche con Burke ratio si ottengono risultati migliori per $L = 5$, in particolare con $\varepsilon = \{10\%, 15\%\}$, invece ottiene risultati peggiori con

N=1	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	L	ε (%)	G €	N €	G %	N %	G %	N %	G %	N %	#
SR	5	5	88,432	79,084	-0,003	-0,006	-0,764	-1,454	28,189	13,203	9,3
		10	138,200	126,953	0,008	0,006	2,040	1,501	80,134	77,632	7,1
		15	141,594	127,951	0,009	0,006	2,195	1,550	77,731	77,211	8,4
	22	5	80,956	72,056	-0,005	-0,008	-1,310	-2,025	46,544	39,014	9,7
		10	92,517	78,730	-0,002	-0,006	-0,484	-1,482	64,082	42,482	13,4
		15	72,543	59,456	-0,008	-0,013	-1,984	-3,193	71,712	68,863	16,5
BR	5	5	86,262	78,429	-0,004	-0,006	-0,918	-1,505	3,493	3,344	7,9
		10	141,080	126,723	0,009	0,006	2,172	1,489	79,713	77,632	8,9
		15	144,883	130,314	0,009	0,007	2,341	1,667	76,889	76,393	8,8
	22	5	69,016	57,435	-0,009	-0,014	-2,288	-3,402	5,673	3,369	15,3
		10	82,289	70,040	-0,005	-0,009	-1,209	-2,198	74,090	61,506	13,4
		15	61,335	48,127	-0,012	-0,018	-3,005	-4,462	7,852	1,486	20,2
SOR	5	5	175,480	164,414	0,014	0,012	3,573	3,152	80,852	79,688	5,4
		10	164,948	153,144	0,012	0,011	3,173	2,696	89,720	89,671	6,2
		15	115,520	100,087	0,004	0,000	0,905	0,005	89,200	82,190	11,9
	22	5	62,834	54,490	-0,012	-0,015	-2,859	-3,719	20,585	20,064	11,9
		10	100,258	93,012	0,000	-0,002	0,016	-0,451	57,468	28,759	6,2
		15	84,939	75,553	-0,004	-0,007	-1,014	-1,735	17,364	16,200	9,7

Tabella 5.17: : HERA S.p.A. per QL con N=1

$L = 22$ soprattutto $\varepsilon = 15\%$, dove la percentuale over 100 risulta pari a 1,486%. Sortino in generale ottiene i risultati migliori, soprattutto nei rendimenti per $L = 5$ dove per $\varepsilon \in \{5\%, 10\%\}$ ottiene i rendimenti annui netti più alti della tabella con i valori, rispettivamente, di 3,152% e 2,696%. Sempre negativi tuttavia anche per Sortino i rendimenti per $L = 22$, che però sono migliori rispetto agli altri indici anche se comunque decisamente negativi soprattutto in termini di percentuali over 100. Se si confrontano questi risultati con la tabella 5.18 per $N = 5$, si nota che 11 valori su 18 migliorano, in particolare in Sharpe dove tutti i rendimenti ad eccezione di $L = 5$ e $\varepsilon = 5\%$ migliorano, mentre per Sortino non sembrano esserci evidenti cambiamenti tra $N = 1$ ed $N = 5$. Con Burke ratio invece migliorano le percentuali over 100 piuttosto che i rendimenti. Questo significa che in generale le performance durante l'intero periodo di investimento sono positive "più spesso" rispetto ad $N = 1$. Si potrebbe pensare che, sulla base di quanto visto dai risultati dei precedenti titoli, Burke tende a compiere meno azioni rispetto agli altri indici, quindi il maggior valore di percentuale over 100 potrebbe essere conseguenza della rigidità dell'indice Burke. Per osservare più attentamente

N=5	QL		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	L	ε (%)	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	#
SR	5	5	70,947	62,961	-0,009	-0,011	-2,122	-2,849	17,183	16,365	9,9
		10	181,838	167,300	0,015	0,013	3,807	3,268	76,915	76,519	6,9
		15	155,437	141,509	0,011	0,009	2,794	2,193	79,420	76,891	7,8
	22	5	84,009	76,717	-0,004	-0,007	-1,083	-1,642	27,002	17,034	7,6
		10	97,415	88,820	-0,001	-0,003	-0,164	-0,738	47,731	38,681	7,7
		15	92,593	79,276	-0,002	-0,006	-0,480	-1,441	79,965	73,246	12,9
BR	5	5	56,263	48,829	-0,014	-0,018	-3,530	-4,380	14,605	8,604	11,8
		10	131,179	113,836	0,007	0,003	1,710	0,813	90,379	88,619	11,8
		15	152,651	133,276	0,010	0,007	2,678	1,811	89,412	87,379	11,3
	22	5	90,050	77,517	-0,003	-0,006	-0,653	-1,579	97,868	71,113	12,5
		10	61,576	50,933	-0,012	-0,017	-2,984	-4,128	59,459	17,977	15,8
		15	70,964	61,433	-0,009	-0,012	-2,120	-2,999	92,487	72,080	12,0
SOR	5	5	116,030	106,922	0,004	0,002	0,933	0,419	52,839	48,351	6,8
		10	191,272	168,236	0,016	0,013	4,136	3,304	90,503	88,594	10,7
		15	97,969	84,246	-0,001	-0,004	-0,128	-1,065	76,370	75,527	12,6
	22	5	87,449	75,409	-0,003	-0,007	-0,835	-1,748	92,115	65,708	12,4
		10	84,709	80,606	-0,004	-0,005	-1,032	-1,338	10,141	8,629	4,1
		15	91,687	86,348	-0,002	-0,004	-0,541	-0,913	3,769	3,719	5,0

Tabella 5.18: HERA S.p.A. per QL con N=5

si vedano i seguenti grafici nella Figura 5.26 che rappresentano Burke con QL per $L = 22$ e $\varepsilon = 5\%$, dove rispettivamente sono illustrati i casi $N = 1$ a sinistra e $N = 5$ a destra. Dal confronto tra i due appena osservate, si nota che nel caso $N = 5$ Burke compie meno azioni pari a 12,5 per anno, rispetto alle 15,3 all'anno compiute con $N = 1$. Si nota come la differenza di performance nasce dai primi istanti di tempo t dove in $N = 1$ assume una posizione di vendita mentre con $N =$

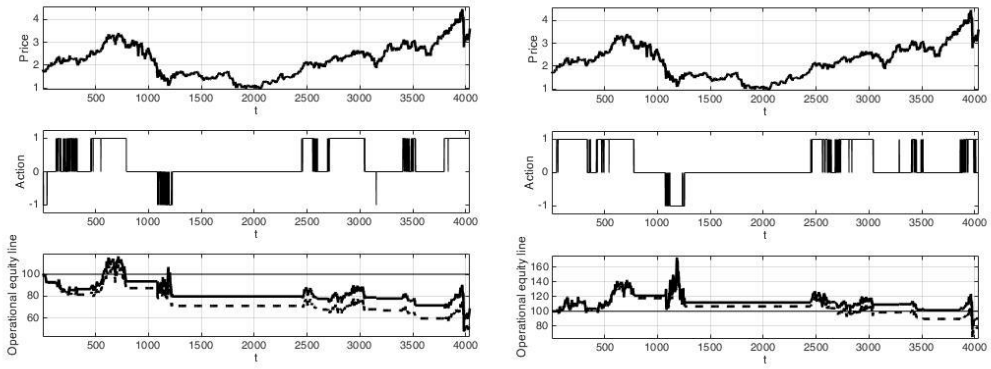

Figura 5.26: HERA S.p.A., Burke ratio QL, L=22, ε = 5%. A sinistra: N=1. A destra: N=5. Fonte: Matlab

5 assume una posizione di acquisto, portando le *equity lines* a due direzioni diverse, il primo in perdita mentre il secondo di guadagno. Anche in questo caso, con più informazioni che descrivono gli stati s_t, Burke è in grado di compiere scelte migliori.

Nelle prossime tabelle vengono presentati i risultati per SARSA. Dalla prima tabella per SARSA con $N = 1$, si nota un leggero miglioramento per Sharpe e Burke. Sharpe in particolare ottiene rendimenti positivi per $L = 22$ e maggiori di quelli ottenuti in QL. Con $L = 5$ i rendimenti sono tutti negativi, solo uno dei quali migliora con SARSA. Tuttavia, le percentuali over 100 peggiorano in tutti i casi. Con Burke solo il valore per $L = 5\ \varepsilon\ = 5\%$ è positivo mentre gli altri sono negativi, anche se con $L = 22$ diventano migliori comunque rispetto a QL. Si noti però che le percentuali associate a questi rendimenti negativi per Burke con SARSA sono pessime, tanto che non superano il 7%. In generale negativo per Sortino che ottiene ora tutti rendimenti negativi anche se due di questi sono comunque maggiori del caso QL. In più, mentre con QL le percentuali per $L = 5$ sono alte, con SARSA peggiorano con un valore massimo di 40%. Dunque, a parte per Sharpe $L = 22$ dove i rendimenti migliorano (anche se le percentuali over 100 peggiorano) in generale negli altri casi non si ottengono risultati migliori con SARSA. Dalla seconda ed ultima tabella, Sharpe risulta migliore con $L = 5\ \varepsilon\ = 5\%$ dove ottiene sia un rendimento positivo a differenza di tutte le altre tabelle, sia una percentuale over di circa il 94%, mentre negli altri casi non superava il 17%. Le altre combinazioni invece risultano in linea o peggiori sia di QL sia di SARSA con $N = 1$. Burke al contrario ottiene delle buone performance rispetto agli altri casi. Per $L = 22$ e $L = 5\ \varepsilon\ = 5\%$ infatti ottiene tutti rendimenti positivi con percentuali over 100 prossime al 100%. Inoltre per $L = 22$, per ogni ε, Burke è l'indice che ottiene le performance migliori della tabella. Per gli altri due casi invece non ottiene né rendimenti positivi né buone percentuali over 100 che non superano il 9%. Sembra quindi che Burke con meno rendimenti non performi peggio, per più rendimenti e maggiori informazioni sugli stati, ottenga performance migliori. Tuttavia il numero di operazioni, soprattutto con $L = 22$ è prossimo allo zero. Sortino invece ottiene rendimenti migliori con SARSA $N = 5$ solo per $L = 22\ \varepsilon\ = \{5\%, 15\%\}$,

N=1	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	86,791	82,223	-0,004	-0,005	-0,880	-1,214	5,227	3,344	4,5
		10	87,344	85,016	-0,003	-0,004	-0,841	-1,008	6,465	4,756	2,2
		15	95,190	92,790	-0,001	-0,002	-0,307	-0,466	8,571	8,447	2,1
	22	5	116,730	111,522	0,004	0,003	0,970	0,683	30,320	26,728	3,8
		10	119,840	117,969	0,004	0,004	1,136	1,037	37,577	35,076	1,3
		15	107,876	105,078	0,002	0,001	0,474	0,310	26,282	24,697	2,2
BR	5	5	112,692	109,863	0,003	0,002	0,749	0,589	91,801	90,587	2,1
		10	77,545	71,404	-0,006	-0,008	-1,575	-2,081	3,369	3,344	6,9
		15	69,256	59,734	-0,009	-0,013	-2,267	-3,165	3,592	3,542	12,3
	22	5	74,854	68,517	-0,007	-0,009	-1,792	-2,332	3,815	0,297	7,4
		10	86,743	81,819	-0,004	-0,005	-0,884	-1,245	9,438	6,911	4,9
		15	70,081	63,088	-0,009	-0,011	-2,195	-2,834	0,347	0,347	8,7
SOR	5	5	85,351	75,081	-0,004	-0,007	-0,984	-1,773	8,323	3,889	10,7
		10	99,630	91,691	0,000	-0,002	-0,023	-0,540	50,904	39,708	6,9
		15	104,664	89,070	0,001	-0,003	0,285	-0,720	86,104	37,800	13,4
	22	5	75,553	74,429	-0,007	-0,007	-1,735	-1,827	17,761	17,092	1,2
		10	84,967	78,468	-0,004	-0,006	-1,012	-1,502	15,432	15,432	6,6
		15	93,330	85,808	-0,002	-0,004	-0,430	-0,951	18,999	17,637	7,0

Tabella 5.19: HERA S.p.A. per SARSA con N=1

N=5	SARSA		Equity line		Rendim giorn		Rendim annuo		perc over 100		num op
rwd	**L**	**ε (%)**	*G €*	*N €*	*G %*	*N %*	*G %*	*N %*	*G %*	*N %*	*#*
SR	5	5	121,977	121,249	0,005	0,005	1,249	1,211	93,975	93,950	0,5
		10	88,805	81,527	-0,003	-0,005	-0,739	-1,268	6,323	3,149	7,1
		15	67,811	53,371	-0,010	-0,016	-2,398	-3,847	7,166	3,794	19,9
	22	5	120,769	117,459	0,005	0,004	1,186	1,011	30,722	26,184	2,3
		10	109,695	105,885	0,002	0,001	0,580	0,358	15,745	13,315	2,9
		15	68,713	56,750	-0,009	-0,014	-2,317	-3,478	19,291	19,142	15,9
BR	5	5	120,419	118,624	0,005	0,004	1,168	1,073	91,346	91,198	1,2
		10	76,333	67,543	-0,007	-0,010	-1,673	-2,422	76,221	8,902	10,2
		15	64,332	51,468	-0,011	-0,016	-2,719	-4,065	5,207	5,033	18,6
	22	5	117,997	117,821	0,004	0,004	1,039	1,030	99,752	99,727	0,1
		10	109,322	108,017	0,002	0,002	0,558	0,483	99,207	98,934	1,0
		15	120,686	116,587	0,005	0,004	1,182	0,964	98,165	97,992	2,9
SOR	5	5	106,287	101,223	0,002	0,000	0,382	0,076	93,528	90,677	4,1
		10	130,173	118,539	0,007	0,004	1,661	1,068	98,611	97,868	7,8
		15	88,096	77,964	-0,003	-0,006	-0,789	-1,543	33,647	22,043	10,2
	22	5	79,088	75,720	-0,006	-0,007	-1,455	-1,723	17,927	15,696	3,6
		10	84,253	80,304	-0,004	-0,005	-1,065	-1,361	15,423	13,861	4,0
		15	105,971	97,441	0,001	-0,001	0,363	-0,162	98,438	17,952	7,0

Tabella 5.20: HERA S.p.A. per SARSA con N=5

tuttavia le performance over 100 sono molto basse indicando una performance in generale negativa per quasi tutto il periodo di investimento. Migliorano invece i valori per $L = 5\ \varepsilon = \{5\%, 10\%\}$ con $N = 5$, ai quali si associano percentuali prossime al 100.

Si vedano le performance di Burke con SARSA $N = 5$ nella prossima figura. A sinistra si rappresenta il caso di $\varepsilon = 5\%$, a destra $\varepsilon = 15\%$. Si nota come torna anche in questo caso la rigidità dell'indice di Burke che compie delle azioni solo nel primo periodo, poi rimane fuori dal mercato fino al termine dell'investimento. Il risultato è lo stesso sia per ε piccoli che grandi, dunque il maggior valore di *exploration* non è comunque sufficiente a rendere l'algoritmo con Burke più reattivo ai movimenti dei prezzi. Se si confronta questa figura con il grafico di destra nella precedente Figura 5.26 si nota che c'è una notevole differenza tra le azioni compiute con QL: prima sono tante e lungo tutto il periodo di investimento, mentre con SARSA sono poche e solo nel primo periodo.

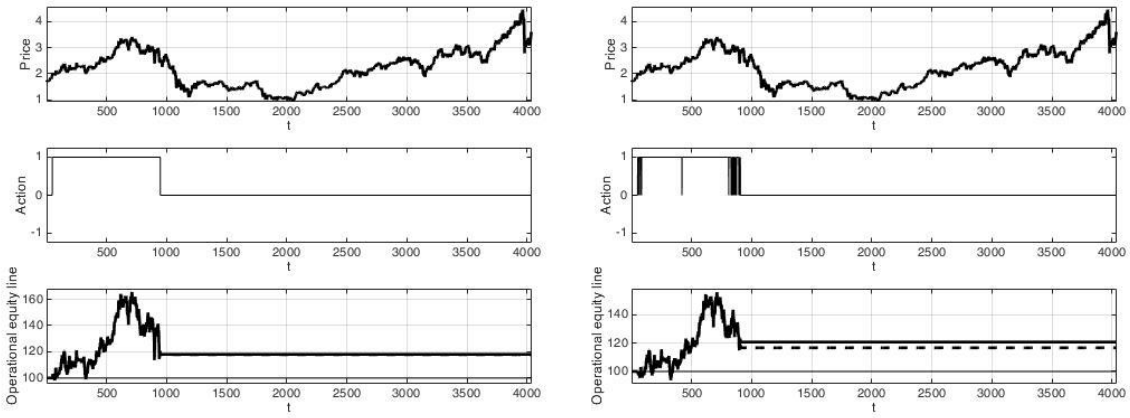

Figura 5.27: HERA S.p.A., Burke ratio SARSA, L=22, $N = 5$. A sinistra: $\varepsilon = 5\%$. A destra: $\varepsilon = 15\%$. Fonte: Matlab.

5.2 Osservazioni

Dall'analisi compiuta sui singoli titoli emergono alcune osservazioni dalle quali si possono trarre le conclusioni di questa applicazione.

I primo risultato è che le prestazioni degli algoritmi e delle funzioni di *reward* dipendono dall'andamento dei prezzi del titolo che si sta esaminando. Come è stato osservato lungo il capitolo (soprattutto evidenziato dai grafici a tre pannelli), in periodi di trend incerti si sono realizzate delle performance negative in generale per tutte le configurazioni dei due algoritmi, indipendentemente dal settaggio dei parametri. Invece, in periodi di trend ben definiti, gli algoritmi ottengono delle indicazioni più esplicite, che si riflettono in un apprendimento più efficace e quindi migliori performance delle strategie di trading. Per esempio, si osservi la seguente Figura (5.28) che illustra nel primo pannello l'andamento del prezzo del titolo Campari S.p.A. e negli altri tre pannelli le *equity lines* ottenute con QL, rispettivamente, per Sharpe, Burke e Sortino ratio. Si può vedere come nel primo periodo di investimento i prezzi oscillino senza una tendenza definita, cioè in condizioni di cosiddetto mercato laterale, portando le *equity lines* a dei rendimenti negativi fino a circa l'istante $t = 3000$. Da questo instante in poi l'apprendimento fa sì che i segnali operativi che derivano dal trend positivo dei

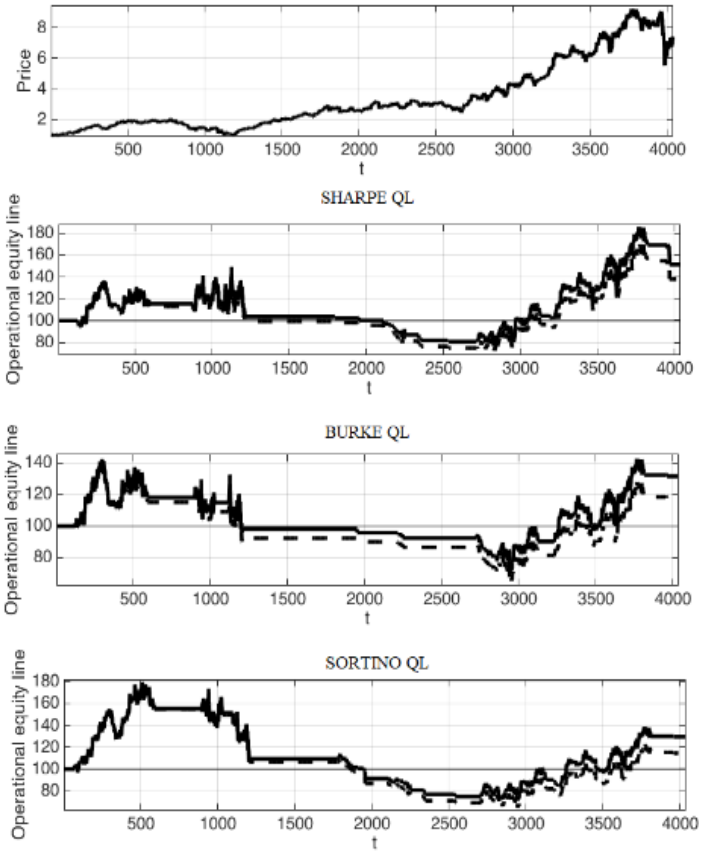

Figura 5.28: In alto: andamento dei prezzi del titolo Campari S.p.A. Negli altri pannelli: equity lines, QL, rispettivamente, per Sharpe, Burke e Sortino. Parametri: $L = 5.\ N = 1.\ \varepsilon = 10\%$.

prezzi vengano tradotti in modo efficiente da tutte le funzioni di *reward*, portando ad operazioni profittevoli man mano che l'investimento avanza. Quindi, l'andamento dei prezzi influenza l'apprendimento degli algoritmi: si deve tenere in considerazione che in condizioni di mercato laterale c'è la possibilità che si creino una serie di falsi segnali che portano a delle operazioni di investimento sbagliate. Per evitare questo problema si suggerisce di modificare il *learning rate* α_t che influisce sulla velocità di apprendimento dell'algoritmo, aumentandone il suo valore. Nel settare α_t si deve tenere in considerazione che ci sono dei i criteri di convergenza da rispettare (descritti nel Capitolo 3), senza i quali non si riuscirebbe ad ottenere l'ottimo della funzione valore. Per evitare il problema, un'alternativa all'aumento del parametro α_t potrebbe essere quello di introdurre due diversi learning rate settaggio di due valori diversi di α_t, da applicare a seconda dell'ambiente di riferimento: uno abbastanza grande da applicare agli algoritmi soltanto durante i periodi di mercato in fase laterale così da essere in grado di superarli senza eccessiva difficoltà nella traduzione dei segnali operativi, uno abbastanza piccolo per assicurare la convergenza durante il resto dei periodi. Si lascia questa alternativa come spunto per un approfondimento futuro.

Le performance analizzate nei paragrafi precedenti vengono riassunte nella tabella 5.21 che riporta la media dei rendimenti annui netti in percentuale per i tre

Media rendim annui netti con QL (%)					
	Amplifon	Azimut	Banco BPM	Campari	HERA
Sharpe	5,417	0,443	-0,326	0,908	-0,526
Burke	2,823	3,126	0,602	-1,091	-1,573
Sortino	2,339	2,240	0,563	-1,298	-0,116

Media rendim annui netti con SARSA (%)					
	Amplifon	Azimut	Banco BPM	Campari	HERA
Sharpe	-0,439	-0,111	-5,434	2,717	-0,556
Burke	-1,282	2,639	-1,154	1,485	-1,167
Sortino	0,796	0,384	-1,404	1,078	-0,913

Tabella 5.21: Media dei rendimenti percentuali annui netti ottenuti dai risultati delle tabelle precedenti. Elaborazione in Excel.

ratio, divisa tra QL (sopra) e SARSA (sotto). I dati riportati sono il risultato delle medie dei rendimenti per i vari parametri (L, N, ε). Indipendentemente dai settaggi l'algoritmo QL performa meglio di SARSA per tutti i titoli, ad eccezione di Campari S.p.A. che ottiene ottimi valori di performance con SARSA (per tutti e tre i ratio). Questa eccezione, nonostante gli ottimi risultati, è la conseguenza di una strategia di investimento praticamente immobile di SARSA, cioè una strategia composta da pochissime operazioni di trading che si concentrano principalmente all'inizio del periodo, mentre nella restante e maggior parte del periodo resta fuori dal mercato (si veda Figura 5.23, 5.24, 5.25). SARSA effettivamente realizza ottimi rendimenti rispetto a QL nel caso di Campari S.p.A. però il modo in cui li ottiene mostra delle difficoltà a livello operativo. Al contrario, QL risulta un algoritmo più operativo nel mercato, cioè in grado di valutare l'andamento del prezzo di un titolo e di assumere delle posizioni nel mercato portando a strategie profittevoli. A conferma di questo si osservi anche la Tabella 5.22 che riassume il numero di operazioni annue medie compiute dai due algoritmi nei cinque titoli. Dai dati si vede come le operazioni annue compiute siano in media il doppio per QL rispetto a SARSA, confermando la minore sensibilità di quest'ultimo. Considerando che i mercati finanziari sono caratterizzati da un certo grado di dinamismo e velocità di risposta agli eventi, si può concludere che SARSA appare meno adatto rispetto a QL ad operare nei mercati finanziari. Infine, con qualche calcolo si può vedere che i rendimenti al termine dell'investimento sono positivi o uguali a zero nel 62% dei risultati per QL, nel 51% per SARSA. Invece, le percentuali over 100 sono positive[35] per QL nel 70% dei casi mentre per SARSA solo nel 47%, a conferma di quanto appena osservato.

Media operazioni annue						
	Amplifon	Azimut	Banco BPM	Campari	HERA	Media
QL	9,7	8,7	7,4	10,9	10,3	9,4
SARSA	5,3	2,8	6,4	3,3	6,2	4,8

Tabella 5.22: Media delle operazioni annue per QL e SARSA. Elaborazione in Excel.

[35] Si ricorda che vengono considerati positivi i valori di percentuale over 100 maggiori del 50%, indicando che nel 50% del tempo di investimento l'equity line è almeno pari al capitale investito.

Per quanto riguarda il confronto tra le funzioni di *reward*, i risultati hanno mostrato che il Sortino ratio si comporta in modo molto simile allo Sharpe ratio, in maniera evidente soprattutto quando si confrontano i grafici a tre pannelli. In questi grafici si è visto come sia il pannello delle azioni intraprese sia il pannello dell'*equity line* risultante fossero molto simili. Questo suggerisce quindi che gli algoritmi recepiscono i segnali operativi e intraprendono strategie di investimento in modo simile. Elaborando i risultati, si ottengono i seguenti dati (Tabella 5.23) che indagano le differenze di performance ottenute con Sharpe e quelle ottenute con Sortino, in termini di quante volte Sharpe ottiene rendimenti (in alto) e percentuali over 100 (in basso) più alti rispetto a Sortino. Guardando ai risultati delle tabelle, Sharpe è più efficiente nella scelta delle strategie di investimento con QL rispetto che con SARSA, Sortino invece sembra ottenere risultati migliori rispetto a Sharpe quando utilizza l'algoritmo SARSA. Tuttavia, i valori in media si aggirano intorno al 50% in tutti i quattro casi. Quindi si può concludere che Sharpe ratio e Sortino ratio ottengono performance non distanti in termini di efficienza, e che con QL si consiglia di utilizzare Sharpe, mentre con SARSA si consiglia Sortino.

Sharpe vs Sortino - confronto rendimenti realizzati						
	Amplifon	Azimut	Banco BPM	Campari	HERA	Media
QL	75%	33%	29%	71%	42%	50%
SARSA	33%	42%	17%	46%	63%	40%

Sharpe vs Sortino - confronto % over 100 realizzate						
	Amplifon	Azimut	Banco BPM	Campari	HERA	Media
QL	67%	25%	63%	54%	50%	52%
SARSA	54%	42%	46%	46%	50%	48%

Tabella 5.23: In alto: le percentuali di volte in cui Sharpe ha ottenuto rendimenti maggiori di Sortino. In basso: le percentuali di volte in cui Sharpe ha ottenuto %over100 maggiori di Sortino. Elaborazione in Excel.

Passando ad analizzare l'indice di Burke, in precedenza si è osservato che le traiettorie delle *equity lines* e, in particolare, i pannelli raffiguranti le posizioni assunte nei mercati sono diverse rispetto a Sharpe. Burke risulta il meno performante dei tre ratio, poiché mostra una bassa sensibilità operativa alle oscillazioni dei prezzi, portando più spesso a preferire $a_t = 0$ ("uscire – o restare

fuori – dal mercato"). Questa rigidità deriva dalla struttura del denominatore che caratterizza l'indice di Burke: solo i movimenti negativi dei prezzi vengono considerati sotto forma di *drawdown*, mentre i movimenti positivi vengono considerati pari a 0, portando l'intero indice al valore di 0. Come conseguenza, la struttura del Burke ratio porta un reward pari a 0 (o prossimo) in modo più frequente rispetto agli altri ratio, annullando la sua funzione di misura di performance. Questo comportamento caratterizza la maggior parte dei risultati ottenuti con Burke ratio, e lo rende meno idoneo rispetto agli altri ratio. Per valutare l'efficienza strategica a confronto con Sharpe si osservano in Tabella 5.24 le percentuali di volte in cui Sharpe ha ottenuto rendimenti (tabella in alto) e percentuali over 100 più alte rispetto a Burke. Guardando i rendimenti nella tabella in alto, diversamente da prima, le percentuali sono diverse al variare del titolo, passando da un minimo di 17% a un massimo di 92%. Si può dedurre che Burke e Sharpe ottengono risultati meno "in linea" tra loro, dunque che operativamente Burke ratio interpreta i segnali operativi diversamente rispetto a Sharpe, al contrario di Sortino. Le percentuali over 100 mostrate dalla Tabella 5.24 invece mostrano che nella maggior parte dei casi[36] Burke performa in modo più efficiente per tutto l'arco del periodo di investimento. Come Sortino, anche Burke appare più adatto ad essere utilizzato con SARSA piuttosto che QL. I risultati che si ottengono con questa funzione di *reward*, anche se positivi e

Sharpe vs Burke - confronto rendimenti realizzati						
	Amplifon	Azimut	Banco BPM	Campari	HERA	Media
QL	92%	17%	33%	83%	79%	61%
SARSA	67%	8%	0%	54%	75%	41%

Sharpe vs Burke - confronto % over 100 realizzate						
	Amplifon	Azimut	Banco BPM	Campari	HERA	Media
QL	100%	33%	54%	75%	54%	63%
SARSA	79%	38%	58%	42%	54%	54%

Tabella 5.24: In alto: le percentuali di volte in cui Sharpe ha ottenuto rendimenti maggiori di Burke. In basso: le percentuali di volte in cui Sharpe ha ottenuto %over100 maggiori di Burke. Elaborazione in Excel.

[36] Ad eccezione di Azimut S.p.A. per entrambi gli algoritmi e Campari S.p.A. per SARSA.

migliori rispetto agli altri ratio, sono dovuti al fatto che spesso mette in atto delle strategie solo nel primo periodo, per poi restare fuori dal mercato fino al termine dell'investimento, soprattutto quando l'algoritmo in uso è SARSA. La media delle operazioni annue compiute da Burke con SARSA è infatti pari a 3,9 contro una media di 4,5 per Sortino e 6,0 per Sharpe. Si può quindi concludere che il Burke ratio non si presti ad essere una funzione di *reward* più idonea rispetto a Sharpe, dato che resta "fuori dal gioco" spesso e per tempi lunghi senza intraprendere strategie di investimento. Viene a mancare in questo modo l'obiettivo dell'implementazione degli algoritmi, per i quali si vuole rendere l'agente indipendente e capace di apprendere e affinare strategie di trading finanziario reagendo alle dinamiche dei prezzi dei mercati. Per cercare di ovviare a questo problema operativo che lo caratterizza, si potrebbe pensare di modificare gli intervalli che definiscono i segnali operativi. Infatti, si potrebbe per esempio ridurre la finestra associata all'azione $a_t = 0$, con il seguente sistema:

$$a_t = \begin{cases} -1 < \bar{a}_t < -\frac{1}{6} \; allora \; a_t = -1 \; posizione \; di \; vendita \; o \; "short" \\ -\frac{1}{6} \leq \bar{a}_t \leq \frac{1}{6} \;\; allora \; a_t = 0 \; stare \; fuori \; dal \; mercato \\ \frac{1}{6} < \bar{a}_t < +1 \; allora \; a_t = +1 \; posizione \; di \; acquisto \; o \; "long" \end{cases}$$

Analizzando il settaggio del parametro N, risultano dei miglioramenti con $N = 5$: su 90 rendimenti annui netti totali (18 combinazioni per titolo), circa il 63% dei rendimenti[37] sono migliori con $N = 5$. Risulta quindi che quando vengono utilizzate più informazioni sul passato si ottengono prestazioni migliori, nonostante il maggior numero di elementi del vettore s_t porti a uno sforzo computazionale maggiore. Questa percentuale vale anche quando si considerano i ratio separatamente, per cui si può dedurre che $N = 5$ è consigliabile rispetto ad $N = 1$ indipendentemente dal ratio e dall'algoritmo in uso.

[37] Più precisamente, sono migliori in 56 combinazioni su 90 per QL, 58 combinazioni su 90 per SARSA. I rendimenti a cui si fa riferimento sono quelli mostrati dalle varie tabelle lungo il paragrafo 5.1.

Passando ad analizzare cosa accade al variare del parametro L si riporta la Tabella 5.25 con i rendimenti percentuali medi annui ottenuti al variare di L ed N per QL[38]. Dai dati si evince che in generale $L = 5$ performa meglio quando si applica a Burke ratio e Sortino ratio, mentre con Sharpe ratio non c'è un valore di L che porta a risultati migliori dell'altro: per $N = 1$ le performance migliori si ottengono se $L = 22$, con $N = 5$ invece sembra preferirsi $L = 5$, in linea con gli altri ratio. Lo stesso risultato si ottiene con SARSA.[39] In generale quindi, entrambi gli algoritmi riescano ad ottenere strategie di investimento più efficienti quando, usando i ratio Burke e Sortino, la scelta della posizione da assumere nel mercato si basa sui rendimenti dell'ultima settimana di trading, piuttosto che sull'ultimo mese. Rimane ambiguo invece il comportamento con Sharpe, risultato che suggerisce di compiere un approfondimento sul comportamento del parametro L con Sharpe ratio per QL, come si vedrà nel prossimo paragrafo dove si prova a testare il comportamento di $L = 10$.

Confronto dei rendimenti medi annui al variare di L													
QL		Amplifon		Azimut		Banco BPM		Campari		Hera		Media	
rwd	L	N=1	N=5	N=1	N=5	N=1	N=5	N=1	N=5	N=1	N=5	N=1	N=5
SR	5	4,30	5,06	0,22	0,39	1,02	1,29	0,91	1,96	0,53	0,87	1,16	1,60
	22	5,80	6,50	0,37	0,78	-4,12	0,50	1,56	-0,81	-2,23	-1,27	0,23	0,95
BR	5	1,16	1,21	2,06	2,12	2,14	2,71	-0,53	-0,62	0,55	-0,59	0,90	0,81
	22	4,24	4,67	1,64	6,69	-0,87	-1,48	-0,23	-2,98	-3,35	-2,90	0,24	0,67
SO	5	4,45	4,88	1,37	1,71	1,27	1,71	-1,08	-0,13	1,95	0,89	1,33	1,51
	22	-0,13	0,16	1,90	3,98	-0,32	-0,41	-3,07	-0,91	-1,97	-1,33	-0,60	0,25

Tabella 5.25: Rendimenti % medi annui ottenuti con QL al variare di L, N. Elaborazione in Excel.

Come ultima osservazione, si valuta se ci sono miglioramenti nell'uso dei diversi settaggi del parametro $\varepsilon \in \{5\%, 10\%, 15\%\}$, cioè del grado di esplorazione di azioni scelte in modo casuale che permettono agli algoritmi di sperimentare delle azioni che altrimenti potenzialmente non avrebbero scelto, così da accrescere l'esperienza e l'apprendimento dell'agente sulle possibili strategie e relative conseguenze. Come già anticipato, bisognerebbe determinare un grado di esplorazione tale che permetta di scoprire nuove possibilità, senza però essere così

[38] I valori sono il risultato delle medie dei rendimenti mostrati lungo il paragrafo 5.1 al variare di ε.
[39] Per semplicità si omette la tabella dei risultati.

alto da lasciare la scelta della strategia al caso troppo spesso. Dai risultati ottenuti, i rendimenti sono maggiori con $\varepsilon = 10\%$ nel 64% dei rendimenti rispetto a $\varepsilon = 5\%$, nel 62% rispetto a $\varepsilon = 15\%$. SARSA in particolare è l'algoritmo che ottiene più vantaggi dal settaggio $\varepsilon = 10\%$. A conferma quindi di quanto appena detto, $\varepsilon = 10\%$ risulta il miglior compromesso tra i valori di esplorazione testati portando una performance migliore dell'investimento.

Si conclude per il momento che la miglior combinazione possibile si realizza quando viene applicato l'algoritmo QL come sistema automatico di trading, con l'uso dello Sharpe ratio come misura di performance e il seguente settaggio di parametri: $N = 5, L = 5, \varepsilon = 10\%$. Nel prossimo paragrafo si commentano brevemente alcuni approfondimenti su quanto evidenziato con lo scopo di individuare ulteriori miglioramenti nei settaggi oppure confermare quanto già è stato tratto.

5.3 Altri approfondimenti

Si è visto che il valore del parametro N porta a performance migliori quando è pari a 5 piuttosto che 1, cioè quando le condizioni dell'ambiente ad un istante t è descritto dai rendimenti degli ultimi 5 giorni piuttosto che dal rendimento del giorno precedente. Alla luce di questo risultato, si può approfondire il comportamento del trade-off che si realizza quando aumenta il valore di N dovuto al fatto che, da un lato, l'algoritmo si serve di più informazioni (rendimenti) che descrivono l'ambiente ma, dall'altro lato, le informazioni più distanti nel tempo potrebbero essere fuorvianti per interpretare movimenti dei prezzi più recenti, rendendo l'algoritmo poco reattivo. Si è ritenuto utile approfondire il comportamento quando $N = 3$, cioè quando a descrivere il vettore s_t si utilizzano i rendimenti degli ultimi 3 giorni di trading, utilizzando QL con Sharpe ratio, $L = 22$ e $\varepsilon = 10\%$. I risultati si riassumono in Tabella 5.26, dove si raccolgono i risultati per Sharpe quando con QL vengono settati, in ordine, $N = 1,3,5$. Mentre in precedenza si otteneva che il 63% dei rendimenti erano maggiori con $N = 5$, ora il confronto tra i tre settaggi mostra che i risultati sono maggiori nel 27% dei casi con $N = 1$, nel 33% dei casi con $N = 3$ e nel 40% dei casi con

Confronto rendimenti medi annui al variare di N					
L	AMPL	AZIMUT	BANC	CAMP	HERA
N=1					
5	2,21	-0,40	4,12	0,13	-1,45
	5,25	-0,70	-2,82	2,03	1,50
	5,45	1,77	1,75	0,58	1,55
22	3,98	-2,89	-3,38	2,57	-2,02
	8,87	0,86	-3,42	3,81	-1,48
	4,56	3,16	-5,55	-1,70	-3,19
N=3					
5	3,91	-2,46	4,12	2,36	-2,64
	6,51	0,46	-0,97	0,24	2,35
	5,62	-2,30	3,39	3,18	1,68
22	5,85	-4,72	-3,52	4,53	-3,64
	7,89	0,10	-1,97	-1,75	-0,92
	8,60	1,87	1,54	-1,92	-1,70
N=5					
5	3,84	-1,87	3,55	1,86	-2,85
	4,73	0,37	-1,15	2,74	3,27
	6,62	2,68	1,47	1,29	2,19
22	7,12	-1,16	2,05	3,12	-1,64
	6,37	0,56	0,56	0,33	-0,74
	6,00	2,94	-1,11	-5,87	-1,44

Tabella 5.26: Rendimenti medi annui realizzati al variare del parametro n con QL e Sharpe ratio. In ordine per: $N = 1, N = 3, N = 5$. Elaborazione in Excel.

$N = 5$. Confrontando invece direttamente il caso $N = 3$ con $N = 5$ non c'è un valore che performa in modo nettamente migliore dell'altro, per cui si suggerisce di fare ulteriori approfondimenti. Sulla base di questi risultati, $N = 5$ si conferma il settaggio migliore, confermando, come concluso in precedenza, che si ottengono strategie di investimento più profittevoli quando il vettore che descrive lo stato contiene le informazioni sui rendimenti dell'ultima settimana di trading.

Il secondo approfondimento realizzato valuta cosa accade quando si imposta un valore del parametro L intermedio rispetto a quelli settati nelle prove precedenti, cioè pari a $L = 10$. I risultati precedenti hanno suggerito che in generale i rendimenti maggiori si ottengono quando le misure di performance si basano su un periodo più corto dei rendimenti passati (5 giorni di trading), mentre un periodo più lungo (22 giorni di trading) sembra influenzi negativamente sulla scelta della strategia di investimento perché considera un periodo troppo lontano e

quindi rendimenti che non rispecchiano più gli attuali movimenti dei prezzi. Sharpe invece non ha evidenziato una preferenza. Di conseguenza risulta ora interessante indagare se un valore intermedio tra 5 e 22 giorni di trading del settaggio di L possa indicare in modo più chiaro come influisce sulle performance utilizzando lo Sharpe ratio con l'algoritmo QL. Nella tabella 5.27 si riportano i risultati dei rendimenti medi annui in percentuale per i tre valori di L. Si osserva che quando si considera anche $L = 10$ si ottiene il 50% dei casi migliori per $L = 5$ mentre, rispettivamente, il 30% e 20% con $L = 10$ e $L = 22$. Questo conferma la tendenza generale per cui è preferibile considerare meno rendimenti per il calcolo delle misure di performance.

Confronto dei rendimenti medi annui al variare di L										
QL	Amplifon		Azimut		Banco BPM		Campari		Hera	
L	N=1	N=5	N=1	N=5	N=1	N=5	N=1	N=5	N=1	N=5
5	4,30	5,06	0,22	0,39	1,02	1,29	0,91	1,96	0,53	0,87
10	8,12	9,17	-0,92	0,35	-3,11	0,45	1,76	0,56	-0,23	-0,47
22	5,80	6,50	0,37	0,78	-4,12	0,50	1,56	-0,81	-2,23	-1,27

Tabella 5.27: Rendimenti medi annui realizzati al variare del parametro L con QL e Sharpe ratio. Elaborazione in Excel.

5.4 Risultati finali

Alla luce di quanto si è osservato dai primi risultati e dagli approfondimenti compiuti successivamente, si possono quindi trarre le seguenti conclusioni:

- Tra i due algoritmi, QL è più adatto di SARSA ad operare nei mercati finanziari. SARSA si è mostrato meno operativo ed efficiente, mentre QL risponde meglio al dinamismo tipico dei mercati finanziari, che richiede un certo grado di attività e velocità nella risposta agli eventi. Entrambi mostrano delle difficoltà ad interpretare i segnali operativi in fasi di mercato senza trend, per cui si lascia come spunto l'idea di approfondire il comportamento degli algoritmi al variare dello *step-size parameter* α_t per ovviare al problema.

- Sharpe ratio non ottiene in assoluto delle performance migliori di Sortino, tuttavia risulta più performante quando è applicato all'algoritmo QL, mentre Sortino ratio performa meglio con SARSA. Sulla base di questa osservazione e del punto precedente, QL associato allo Sharpe ratio è consigliabile rispetto a SARSA associato al Sortino ratio.

- Anche il Burke ratio si presta meglio ad essere utilizzato con SARSA piuttosto che QL. Tuttavia il problema centrale di Burke ratio è che resta fuori dal mercato per periodi lunghi senza intraprendere strategie di investimento. Viene a mancare quindi la capacità dell'agente di apprendere e affinare strategie di trading finanziario reagendo alle dinamiche dei prezzi dei mercati. Si suggerisce di testare cosa accade se, al fine di ovviare a queste performance stagnanti, si modifica il sistema che definisce i segnali operativi riducendo l'intervallo di valori del segnale "uscire – o restare fuori – dal mercato".

- Nel settaggio dei parametri, si consiglia di utilizzare i rendimenti passati pari ad una settimana di trading (5 giorni) sia come informazioni per descrivere la situazione attuale prima di prendere una decisione sulla posizione da assumere nel mercato (parametro N), sia come informazioni su cui basare il calcolo della misura di performance e, dunque, sulla scelta effettiva della posizione da assumere nella strategia di trading (parametro L).

- Infine, il grado di esplorazione del sistema automatico di apprendimento risulta ottimale quando $\varepsilon = 10\%$, mostrandosi un buon valore per avere una scelta casuale dell'azione che permetta al sistema di migliorare il suo apprendimento, senza però lasciare troppa casualità nella costruzione di una strategia di investimento.

Capitolo 6

Conclusioni

In questo elaborato si presentano e applicano alcuni sistemi di trading finanziario automatizzato basati su un approccio auto-adattivo di apprendimento automatico conosciuto come *Reinforcement Learning* (RL), del quale i più noti algoritmi Q-*Learning* e SARSA vengono implementati in ambiente Matlab con l'obiettivo di verificare la valenza della recente teoria dei mercati adattivi, che concilia l'ipotesi dei mercati efficienti – tanto nota quanto discussa – con le moderne teorie proposte dalla finanza comportamentale. Il vantaggio principale dell'uso dei sistemi di trading è quello di eliminare per quanto possibile la componente psicologica e implementare sistemi che siano in grado di cogliere le inefficienze che si creano nei mercati finanziari, creando delle strategie di investimento profittevoli in tempo reale.

Questo elaborato ha mostrato innanzitutto che applicando questi sistemi di trading a cinque titoli finanziari si possono realizzare guadagni sull'investimento iniziale, soprattutto utilizzando Q-*Learning*. L'applicazione inoltre propone due indici di performance alternativi allo Sharpe ratio, ossia il Sortino ratio e il Burke ratio. A differenza di Sharpe, che è la misura di performance più utilizzata in letteratura per questi sistemi di trading, le alternative proposte considerano come rischio rapportato al rendimento soltanto i movimenti negativi dei prezzi. Sebbene, in teoria, considerare il rischio come perdite sia più appropriato nei mercati finanziari, dai risultati si è visto che a livello operativo si presentano poco sensibili ai movimenti dei prezzi, in particolar modo il Burke ratio, rendendo le strategie di investimento immobili dato che per la maggior parte del tempo restano fuori dal mercato. Inoltre, entrambi gli indici proposti hanno ottenuto prestazioni migliori quando sono state associate a SARSA, algoritmo meno operativo rispetto a Q-*Learning*. Di conseguenza si evince che la miglior scelta ricade sull'algoritmo

Q-*Learning* con Sharpe ratio come misura di performance. Nel settaggio dei parametri si consiglia di utilizzare i rendimenti passati pari ad una settimana di trading, sia come informazioni per descrivere l'ambiente prima di prendere una decisione sulla posizione da assumere nel mercato (parametro N), sia come informazioni su cui basare il calcolo della misura di performance e, dunque, sulla scelta effettiva della posizione (parametro L). Invece, il miglior grado di esplorazione (parametro ε) che permette al sistema di migliorare il suo apprendimento risulta pari al 10%.

La maggiore difficoltà per entrambi gli algoritmi si è riscontrata quando i mercati si trovano in una fase laterale, poiché si genera la possibilità di falsi segnali che possono portare ad azioni perdenti. Per ovviare a questo problema si lascia come spunto la possibilità di modificare il parametro *step-size* (α) di apprendimento, proponendo due parametri: uno che assicuri le condizioni di convergenza, l'altro grande abbastanza da agevolare l'apprendimento durante le fasi di mercato laterale. Alla difficoltà di operatività riscontrata con Burke ratio si suggerisce la possibilità di modificare gli estremi degli intervalli dei segnali operativi.

Per quanto riguarda gli sviluppi futuri, oltre agli spunti già suggeriti in precedenza, si propone di proseguire la ricerca considerando anche i volumi delle transazioni nel momento in cui si assume una posizione nel mercato e considerando l'introduzione di alcuni indici di analisi tecnica a supporto dei segnali operativi. Infine, nonostante questa applicazione non abbia segnalato valide alternative allo Sharpe ratio, si ritiene sia opportuno continuare la ricerca di nuove funzioni di *reward* o indagare delle soluzioni operative alle alternative proposte, ritenendo che Sharpe ratio sia comunque troppo elementare.

Bibliografia

Barber, B. e Odean, T. (2001). Boys Will Be Boys: Gender, Overconfidence, And Common Stock Investment. *The Quarterly Journal of Economics*. 116. 261-292.

Barto, A.G., Sutton, R.S. (2018). Reinforcement Learning: An Introduction. *The MIT Press*.

Clarke, R. G., Krase, S. and Statman, M. (1994). Tracking Errors, Regret and Tactical Asset Allocation. *Journal of Portfolio Management*. Spring, 16-24.

Corazza, M. e Fasano, G. e Gusso, R. e Pesenti, R. (2019). A Comparison among Reinforcement Learning Algorithms in Financial Trading Systems. *University Ca' Foscari of Venice, Dept. of Economics Research*. Paper Series No. No. 33/WP/2019.

Corazza, M. e Sangalli, A. (2015). Q-Learning and SARSA: a comparison between two intelligent stochastic control approaches for financial trading. *Working Papers 2015:15, Department of Economics, University of Venice "Ca' Foscari", revised 2015.*

Bertoluzzo, F. e Corazza, M. (2012). Testing Different Reinforcement Learning Configurations for Financial Trading: Introduction and Applications. *Procedia Economics and Finance*, 3, 68-77.

Delcey, T. (2019). Samuelson vs Fama on the Efficient Market Hypothesis: The Point of View of Expertise. *Œconomia* , 9-1.

Fama, E. (1965). The Behavior of Stock-Market Prices. *The Journal of Business*, 38(1), 34-105.

Fama, E. (1965). Random Walks in Stock Market Prices. Selected Papers of the Graduate School of Business, University of Chicago, Reprinted in the *Financial Analysts Journal*, September-October 1965; *The Analysts Journal*, London, 1966; *The Institutional Investor*, October 1968, 55-59.

Fama, E. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance, 25*(2), 383-417.

Fama, E. (1998). Market efficiency, long-term returns, and behavioral finance. *The Journal of Financial Economics*, 49(3), 283-306.

Gao, X. e Chan, L. (2000). An algorithm for trading and portfolio management using Q-Learning and Sharpe ratio maximization. *Proceedings of the international conference on neural information processing*, 832-837.

Gervais S., Odean T., (2001). Learning to Be Overconfident, *The Review of Financial Studies*, 14(1), 1–27.

Kahneman, D. e Tversky, A. (1979). Prospect Theory: An Analysis of Decision under Risk. *Econometrica*, Econometric Society, 47(2), 263-291.

Lo, A. W. (2004). The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective. *The Journal of Portfolio Management 30th Anniversary*, 30 (5), 15-2.

Odean, T. (1998), Are Investors Reluctant to Realize Their Losses?. *The Journal of Finance*, 53: 1775-1798.

Samuelson, P. A. (1965). Proof That Properly Anticipated Prices Fluctuate Randomly. *Industrial Management Review, 6*(2), 41-49.

Shefrin, H. e Statman, M. (1985), The Disposition to Sell Winners Too Early and Ride Losers Too Long: Theory and Evidence. *The Journal of Finance*, 40: 777-790.

Simon, Herbert A. (1955). A Behavioral Model of Rational Choice, *The Quarterly Journal of Economics*, 69(1), 99–118.

Singh, S., Jaakkola, T., Littman, M.L., Szepesvri, C. (2000). Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3), 287–308.

Tversky, A., e Kahneman, D. (1981). The framing of decisions and the psychology of choice. *Science*, 211(4481), 453–458.

Werner F. M. De Bondt, e Thaler, R. (1985). Does the Stock Market Overreact? *The Journal of Finance, 40*(3), 793-805.

Wiering, Marco e Van Otterlo, Martijn (Eds.), (2012). Reinforcement Learning: State-of-the-Art. *Springer-Verlag Berlin Heidelberg*.