Quando si parla di mercati finanziari, una delle prime cose a cui si pensa è il trading floor di uno stock exchange, ovvero la borsa. L'immagine che una persona che non lavora in finanza ha di questo luogo è spesso quella che viene trasmessa dai film e da alcune immagini d'archivio: un luogo schizzofrenico in cui i telefoni continuano a squillare e i pit traders gridano e si comunicano a gesti gli ordini dei clienti. Tuttavia, quest'immagine non corrisponde più alla realtà e risulta oggi altamente anacronistica.

A partire all'incirca dalla fine degli anni '80, con l'avvento dei mercati elettronici, il sistema dell' "open outcry", ovvero la borsa alle grida, è progressivamente scomparsa. Oggi gli ordini vengono processati in cluster di server nascosti in edifici di massima sicurezza in località anonime. Si è perciò assistito a una progressiva computerizzazione della finanza e alla diffusione di algoritmi capaci di prendere decisioni finanziarie ad una velocità inimmaginabile per un trader umano. È stato stimato che nel 2014, all'incirca il 75% degli ordini eseguiti sui mercati equity mondiali sia stato generato da un algoritmo. Perciò, gli algoritmi stanno assumendo un ruolo più importante nei mercati finanziari e più in generale sulla nostra vita di tutti I giorni.

In questa tesi si è studiato come ottenere una strategia di trading tramite Reinforcement Learning, una famiglia di metodi di ottimizzazione che sta ricevendo molta attenzione da parte dell'accademia e dell'industria. La presentazione è strutturata come segue: inizierò con l'introdurre il framework tipico e le nozioni di base del Reinforcement Learning. Approfondirò poi una famiglia particolare di algoritmi che va sotto il nome di Policy Gradient. Presenterò poi un'applicazione numerica ad un problema di gestione di portafoglio con costi di transazione. Infine, farò qualche riflessione conclusiva e presenterò alcuni possibili assi di ricerca per il futuro. (2 min)

Ora cercherò di introdurre da un punto di vista intuitivo il framework standard del Reinforcement Learning. Innanzitutto si considera un "ambiente", come ad esempio il mercato finanziario. In questo ambiente agisce un agente che cerca di realizzare un certo compito, raggiungere un certo obiettivo. Per fare ciò, ad un'istante t, l'agente osserva lo stato del sistema e, seguendo una certa strategia pi greco, seleziona l'azione da compiere. Più formalmente, la strategia pi greco è una funzione che associa allo stato del sistema una distribuzione di probabilità sullo spazio delle azioni. Dopo che l'agente ha eseguito l'azione selezionata, lo stato del sistema evolve secondo la probabilità di transizione P. In seguito a questa transizione, l'agente riceve un guadagno numerico che misura in un certo senso la bontà dell'azione selezionata. Questo chiude il meccanismo di feedback tra l'agente e l'ambiente. Tuttavia, notiamo che l'agente non sa se l'azione selezionata è quella corretta, come nel caso del supervised learning. L'obiettivo dell'agente è perciò quello di trovare la migliore strategia possibile tramite l'interazione con l'ambiente, per tentativi.

Prima di proseguire, formalizziamo il problema. Si definisce una funzione valore sugli stati come il valore atteso dei guadagni futuri scontati che l'agente può ottenere a partire dallo stato s seguendo la strategia pi greco. Si introduce poi una funzione valore sulle azioni che rappresenta il valore atteso dei guadagni futuri scontati a partire da un certo stato quando l'agente sceglie una certa azione. Si definiscono poi delle funzioni valore ottime che rappresentano il massimo valore per un certo stato o coppia stato-azione che l'agente può ottenere al variare della sua strategia. Infine, si definisce strategia ottima quella strategia che permette di ottenere il valore massimo per tutti gli stati. Ci accorgiamo che questo problema è esattamente quello trattato dalla teoria del controllo ottimo stocastico a tempo discreto.

Il metodo tradizionale per risolvere questo problema è il seguente: si parte dal modello del sistema, ovvero dalla probabilità di transizione dell'ambiente e dalla funzione guadagno, e si determinano le funzioni valore ottime tramite le equazioni di Bellman. A partire da queste funzioni, si determina poi la strategia ottima. Tuttavia, nella maggior parte dei casi, non è possibile ottenere una soluzione analitica. Il primo motivo è che le equazioni di Bellman sono estremamente complesse da risolvere.

Il secondo motivo è che, per molte applicazioni pratiche, il modello del sistema è troppo complesso o addirittura sconosciuto. Il Reinfocement Learning può essere perciò visto come una generica famiglia di algoritmi che permette di risolvere il problema di controllo sfruttando delle tecniche di approssimazione funzionale e lavorando direttamente sui dati generati dall'interazione tra agente e ambiente. In base alla componente che si decide di approssimare si ottengono diverse tipologie di metodi: I metodi model-based vanno ad approssimare il modello del sistema. I metodi value-based vanno invece ad approssimare le funzioni valore e derivano da queste un'approssimazione della strategia ottima. Infine, I metodi policy-based approssimano direttamente la strategia ottima. È su questi ultimi che ci siamo concentrati in questa tesi. (6 min)

Passiamo ora a descrivere le idee principali degli algoritmi policy gradient. L'idea chiave di questi metodi è quella di approssimare la strategia ottima con una strategia parametrica, in cui I parametri sono scelti in modo da risolvere il seguente problema di ottimizzazione. Questo problema può essere risolto in maniera numerica utilizzando il metodo del gradiente, che prevede di aggiornare I parametri in maniera iterativa seguendo il gradiente della funzione obiettivo. La domanda a cui bisogna ora rispondere è come calcolare il gradiente.

La risposta si trova nel policy gradient theorem, che afferma che il gradiente della funzione valore è dato dal valore atteso del prodotto della funzione valore Q e il likelihood score, ovvero il gradiente del logaritmo della strategia. Supponiamo che la funzione valore Q sia centrata in zero. Allora quando si ottengono dei guadagni positivi, I parametri della strategia della agente vengono aggiornati seguendo la direzione del likelihood score e viceversa. Questo risultato permette di approssimare il gradiente con un approccio Monte Carlo, rimpiazzando il valore atteso con una media su delle traiettorie simulate del sistema. Questo approccio molto semplice soffre però della grande variabilità che si ha nella stima Monte Carlo. Perciò, si utilizzano in genere delle tecniche per ridurne la varianza, come ad esempio l'utilizzo di baseline ottime, di un critico, o del gradiente naturale. Per questioni di tempo, non entrerò nei dettagli di questi approcci.

Un metodo alternativo che si è rivelato molto efficace si chiama policy gradient con esplorazione nello spazio dei parametri, o PGPE. In questo metodo si utilizza un controllore parametrico che associa ad ogni stato del sistema un'azione in maniera deterministica. Si assume però che I parametri del controllore siano estratti da una iper-distribuzione di probabilità p_xi. In questo caso, la ricerca dell'ottimo non si svolge nello spazio dei parametri del controllore, ma nello spazio degli iperparametri xi. Piu formalmente, lo schema iterativo per risolvere il problema di ottimizzazione diventa il seguente, in cui il gradiente è dato dal seguente teorema. Questo risultato è apparentemente simile al precedente, in cui però compare il likelihood score dell'iperdistribuzione p_xi. Il vantaggio di questo metodo è che, quando si va ad approssimare il valore atteso via Monte Carlo, I parametri del controllore sono simulati una sola volta all'inizio dell'episodio e le azioni sono poi selezionate in maniera deterministica a partire dal controllore F. (9 min)

Passiamo ora a un'applicazione numerica dei metodi appena presentati ad un problema di gestione di portafoglio con costi di transazione. In questo caso, l'obiettivo dell'agente è quello di investire il proprio capitale su I titoli disponibili sul mercato in modo da massimizzare I propri guadagni. I guadagni dell'agente possono essere modellizzati secondo la seguente formula, che esprime il log-rendimento del portafoglio. Il primo termine rappresenta il profitto o la perdita derivante dalla variazione dei prezzi dei titoli. Abbiamo denotato con a I pesi del portafoglio e con X il rendimento del titolo. Il secondo termine corrisponde a un costo di transazione proporzionale al nozionale scambiato quando l'agente ribilancia il portafoglio. Il terzo termino rappresenta il costo che l'agente deve sostenere per vedere allo scoperto delle azioni. Infine l'ultimo termine rappresenta un costo di transazione fisso che l'agente paga ogniqualvolta ribilancia il portafoglio. In questa situazione, l'azione dell'agente corrisponde ai pesi del portafoglio. Assumiamo che ad ogni istante l'agente investe tutto il capitale disponibile cosicche I pesi del portafoglio sommino a 1. Infine bisogna

specificare lo stato del sistema. Cosideriamo I rendimenti dei vari titoli sugli ultimi P giorni e la posizione corrente dell'investitore.

Il primo esperimento che è stato condotto è stato quello di applicare gli algoritmi PGPE ed alcune estensioni originali al fine di trovare una strategia di trading su un titolo sintetico. Per generare questi dati, si è assunto che I log rendimenti del titolo seguino una camminata aleatoria con drift stocastico mean-reverting. La strategia impiegata dall'agente è invece data dalla funzione segno applicata alla combinazione lineare dello stato del sistema. Questo grafico mostra la convergenza dei vari metodi quando non si hanno costi di transazione. Vediamo che gli algoritmi PGPE convergono velocemente e I guadagni ottenuti in media sono nettamente positivi. Al contrario, l'algoritmo ARAC, ovvero un semplice algoritmo actor-critic non è in grado di individuare il pattern presente nei dati e I guadagni ottenuti sono all'incirca nulli. Questo grafico mostra invece le performance delle strategie apprese dagli algoritmi in backtest, ovvero su dei dati independenti che non sono stati utilizzati per il training. Osserviamo che le strategie apprese dagli algoritmi PGPE battono tranquillamente il mercato, producento dei profitti cumulati di all'incirca 370% mentre una semplice strategia Buy and Hold produce solamente l'8%.

Analizziamo ora come le strategia apprese variano al variare dei costi di transazione. Intuitivamente ci si aspetta che, all'aumentare dei costi di transazione, la frequenza con cui l'agente cambia la sua posizione o vende il titolo allo scoperto diminuisca. Questo è effettivamente il caso e mostra come questi algoritmi siano in grado di gestire il tradeoff tra profitti e costi di transazione. L'incapacità di adattarsi ai costi di transazione è tipicamente una delle debolezze degli algoritmi di trading basati solamente sulla predizione dei rendimenti futuri. Sembra perciò che questi algoritmi funzionino almeno su dei dati sintetici.

Abbiamo perciò provato ad applicare questi metodi a dei dati storici. La situazione è decisamente più complessa ed in questi caso non è stato possibile trovare dei risultati soddisfacenti. In alcuni casi I metodi non convergono neanche, mentre in altri I metodi convergono ma le strategie apprese non producono risultati soddisfacenti in backtest. Le possibili spiegazioni di questo risultato sono molteplici e valgono per tutti I metodi statistici che cercano di predirre l'andamento dei mercati finanziari. La prima ragione è che nei dati finanziari si ha tipicamento un alto livello di rumore rispetto al segnale sottostante. È perciò molto difficile trovare dei pattern sfruttabili in maniera sistematica. Il secondo motivo è la qualità dei dati utilizzati. È assai improbabile trovare dei pattern nei prezzi giornalieri di un'azione liquida. Sarebbe assai più interessante applicare questi metodi a dati ad alta frequenza, che sono però molto più difficili da ottenere. La terza possibile spiegazione è che la strategia parametrica utilizzata è troppo semplicistica e le variabili predittive scelte non sono abbastanza potenti per catturare dei segnali. Infine, I mercati finanziari sono tipicamente non stazionari, con vari regimi che si alternano nel tempo. Un signale deve perciò essere persistente nel tempo per poter essere sfruttato.

Arriviamo così alla conclusione di questa presentazione. In questa tesi, è stato svolto uno studio approfondito dei metodi policy gradient e di altre tecniche del Reinforcement Learning. Sono stati fatti dei contributi innovativi alla letteratura di questi metodi che per questioni di tempo e di chiarezza espositiva non sono state presentate oggi. Infine abbiamo cercato di applicare queste tecniche per trovare una strategia di trading. Sebbene questi metodi funzionino molto bene su dei dati simulati, non si può dire lo stesso per l'applicazione sui dati storici. Su questo punto si focalizzano gli assi di ricerca per il futuro. La prima direzione è quella di sviluppare una strategia più sofisticata per l'agente, con features più potenti. Per fare ciò, si potrebbe pensare di combinare queste tecniche con il deep learning e le potenti tecniche che offre, come ad esempio recurrent neural networks o LSTM. Infine, essendo il framework del RL altamente versatile, si potrebbe sicuramente cercare di applicare queste tecniche ad altri problemi finanziari.