Project Work \_ OptimalInvestmentWithQLearning

1. Implementazione della classe

Abbiamo creato una classe figlia OptimalInvestmentWithQLearning che estende la classe TimeDependentQLearning. Come variabili della classe abbiamo definito:

* investmentStep, maximumInvestment, minimumWealthValue, maximumWealthValue, wealthStep, timeStep: vincoli (min/max) e passi di discretizzazione per ricchezza, percentuale investibile e tempo;
* constantDrift, constantVolatility: parametri di drift e volatilità dello stock;
* utilityFunction: funzione di utilità;
* random: generatore di numeri casuali di classe Random, utilizzato in seguito per la simulazione stocastica

Abbiamo poi definito il costruttore della classe e attraverso il comando *super* abbiamo fatto riferimento al costruttore della classe madre specificandone il valore specifico delle variabili in input in questo determinato problema:

* rewardsAtStatesAtFinalTime è stato specificato come un array di lunghezza pari al numero di stati e su ciascun stato viene calcolata la funzione di utilità;
* runningRewards è stata definita come una matrice con numero di righe pari al numero di stati e numero di colonne pari al numero di azioni, è stata inizializzata a zero;
* Parametri del metodo: discountFactor, numberOfTimes, numberOfEpisodes, learningRate, explorationProbability

Nel metodo *getNumberOfActions* abbiamo calcolato il numero di azioni possibili con:  
(maximumInvestment/investmentStep) + 1; che restituisce il numero di azioni da 0 alla percentuale massima investibile, discretizzate con passo specifico della discretizzazione scelta. La prima azione possibile è investire zero, dunque il +1 serve per coerenza.

Nel metodo *computePossibleActionsIndices* si restituisce un array di interi contenente gli indici delle azioni possibili se si ha ricchezza individuata da stateIndex (dato in input). Per prima cosa abbiamo calcolato la ricchezza individuata da stateIndex e il numero di azioni complessive, richiamando in questo caso il metodo sopra descritto. Abbiamo poi definito aValue che rappresenta la percentuale massima investibile che ci assicura di non superare il massimo valore ammesso della ricchezza. Abbiamo definito il vettore actions contenente tutte le percentuali possibili da investire da 0 fino al massimo con rispettiva discretizzazione. A partire da actions e filtrandone i valori, abbiamo creato un vettore contenente solo le percentuali minori di aValue; poiché investire più di tale quantità non risulterebbe ottimale. Infine, ne calcoliamo la lunghezza e restituiamo un array di valori tra 0 (incluso) e la lunghezza appena calcolata (escluso).

Nel metodo *generateStateIndex* viene restituito in output l’indice del nuovo stato (rappresentante la nuova ricchezza) che si raggiungerà se inizialmente si ha ricchezza individuata da oldStateIndex e si decide di intraprendere l’azione identificata da actionIndex. Inizialmente calcoliamo la percentuale investita nello stock identificata da actionIndex e la ricchezza disponibile individuata da oldStateIndex. Il tasso di interesse lo ricaviamo in modo indiretto, invertendo la formula del fattore di sconto. Generiamo un numero casuale attraverso la classe random, che genera una nuova variabile aleatoria con distribuzione normale. Esplicitiamo il drift e la volatilità specifiche del problema di Merton. Calcoliamo il valore della nuova ricchezza, con un passo di eulero, in modo particolare seguendo lo schema di eulero-maruyama con equazione:

Facciamo dei controlli per verificare che la nuova ricchezza individuata sia compresa nell’intervallo ammesso dal problema. Discretizziamo il valore della nuova ricchezza in modo tale da trovarne l’indice corrispondente. Facciamo un ulteriore controllo, verificando che l’indice della nuova ricchezza sia compreso tra 0 e l’indice corrispondente alla ricchezza massima.

1. Esecuzione OptimalInvestmentWithQLearningTest

Una volta eseguito il codice OptimalInvestmentWithQLearningTest abbiamo ottenuto i seguenti risultati:

* dal grafico della Value Function si evince che il massimo guadagno atteso aumenta all’aumentare del valore dello stato che rappresenta la ricchezza attuale. La value function rappresenta il prezzo che siamo disposti a pagare in base allo stato in cui ci si trova. La forma del grafico ricorda la funzione , infatti è concava.

Immagine che contiene Diagramma, linea, pendio, diagramma

Descrizione generata automaticamente

* dal grafico dell’Optimal Investment notiamo che l’andamento è decrescente man mano che si aumenta la ricchezza, come ci aspettiamo dato che ci avviciniamo sempre più alla ricchezza massima detenibile (maximumWealthValue). L’Optimal Investment indica l’azione ottimale che possiamo intraprendere per ottimizzare il nostro payoff.

Immagine che contiene testo, schermata, Diagramma, linea

Descrizione generata automaticamente

1. Fase di test

• Test individuo propenso al rischio (constantDrift = 0.5; exponentForFinalRewardFunction = 2):

* la Value Function, come ci aspettavamo, assume una forma convessa data la propensione al rischio dell’individuo. Di riflesso, i valori del massimo guadagno atteso risultano maggiori anche a causa dell’aumento del drift, infatti maggiore drift implica maggiore crescita attesa, quindi valore atteso più alto.

Immagine che contiene Diagramma, linea, pendio

Descrizione generata automaticamente

* l’Optimal Investment, invece, assume un comportamento simile a quello del test iniziale, ma nei valori di ricchezza compresi tra 0 e 5, l’investitore allocherà una percentuale di ricchezza maggiore nell’asset rischioso. L’andamento complessivo rimane decrescente all’aumentare della ricchezza.

Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente

• Test individuo avverso al rischio (interestRate = 0.03; exponentForFinalRewardFunction = 0.3; constantVolatility = 0.9):

* la Value Function presenta maggiore rumore nei valori compresi tra ricchezza 3 e 8, proprio a causa della volatilità dell’asset rischioso alta (constantVolatility = 0.9). Il massimo guadagno atteso risulta minore rispetto ai due casi precedenti. L’avversione al rischio è aumentata rispetto al caso originale poiché l’esponente dell’utilityFunction è diminuito.

Immagine che contiene Diagramma, linea, schermata, pendio

Descrizione generata automaticamente

* l’Optimal Investment presenta valori minori in generale, poiché l’individuo avverso al rischio preferirà investire nell’asset non rischioso (bond) dato che l’interestRate è di 0.03 e non più nullo.

Immagine che contiene testo, Diagramma, linea, schermata

Descrizione generata automaticamente

• Test individuo propenso al rischio (constantDrift = 0.5; exponentForFinalRewardFunction = 2; timeIndex = numerOfTimes – 2):

* la Value Function assume una forma convessa data la propensione al rischio dell’individuo, come nel primo test. Inoltre, negli stati intermedi presenta dei picchi verso il basso.

Immagine che contiene linea, Diagramma, pendio, diagramma

Descrizione generata automaticamente

* l’Optimal Investment ha una distribuzione di valori meno densa dato che siamo al penultimo istante temporale nel quale è possibile investire. In diversi stati l’azione ottimale è non investire perché l’azione ottimale assume valore 0. Anche qui i valori maggiori sono assunti per ricchezza tra 0 e 5, inoltre per ricchezza superiore a 5 il comportamento della funzione Optimal Investment è decrescente.

Immagine che contiene testo, Diagramma, schermata, linea

Descrizione generata automaticamente