

# TEORIA DEI GIOCHI E INTELLIGENZA ARTIFICIALE: STRATEGIE OTTIMALI PER SISTEMI INTELLIGENTI

## Teoria dei giochi ed intelligenza artificiale

La teoria dei giochi indica il netto passaggio dall'economia classica di Adam Smith, Pigou e Vilfredo Pareto a quella moderna.

Per arrivare alla teoria dei giochi è fondamentale il contributo scientifico offerto in tale direzione da Shapley e Shubik nel 1959.

I due formularono che avendo come base un gioco cooperativo ad utilità trasferibile, caratterizzato da:

- Insieme  $N$  di players;
- $F$  caratteristica  $v$  (guadagno associato alle coalizioni)
- Insieme delle partizioni;

Con questo indice si pesa il contributo marginale che dà il giocatore ad una coalizione  $S$ , dando valore preponderante a tutta la grande coalizione.

Pesa il contributo marginale che il giocatore "i" porta ad ogni possibile coalizione in cui potrebbe entrare, per far sì che la coalizione  $S$  sia dominante, e quindi il giocatore sarà fondamentale.

Gli stessi Shapley e Shubik nel 1969 formalizzarono modelli di equilibrio economico generale in termini di gioco cooperativo con il concetto di Core(nucleo), cioè l'insieme delle imputazioni di un determinato gioco cooperativo che rispondono a determinate proprietà (razionalità individuale e di gruppo) costituiscono l'insieme delle soluzioni che tale gioco comporta.

Nel 1963 Debreu e Scarf hanno dimostrato che per vasti mercati economici, l'equilibrio Walrasiano di concorrenza tende a coincidere con il core di Shapley- Shubik.

L'apporto fondamentale fu dato però da Arrow nel 1954, il cosiddetto approccio all'economia astratta, tramite il celebre teorema di Arrow conosciuto anche come teorema dell'elettore medio.

La teoria dei giochi è stata originariamente formulata dal matematico Von Neumann dal 1928 in poi, il quale in seguito grazie anche al contributo scientifico di Morgenstern del 1944 che ha studiato le applicazioni della stessa in campo economico e sociale. I primi studi della teoria dei giochi si basavano su mercati con concorrenza imperfetta, ovvero il cosiddetto oligopolio.

Lo studio con approccio cooperativo si avrà solo in seguito come anticipato, con Shapley-Shubik. Per questo, da un punto di vista scientifico, negli ultimi anni si è dato molto rilievo all'orientamento non cooperativo fino ad indicarlo come fondamento alla Cournot della teoria dell'equilibrio Walrasiano.

La prima esposizione articolata di gioco non cooperativo è dovuta a Nash nel 1950 e nel 1951 il quale coniò la definizione e dimostrò l'esistenza di un equilibrio per n-giocatori. Questo risultato fondamentale trova frequenti applicazioni nella teoria dell'equilibrio economico generale e per lo studio di problemi rientranti nella teoria dell'oligopolio.

La caratteristica fondamentale dei giochi non cooperativi consiste nel fatto che i giocatori non possono stabilire tra di loro accordi attraverso i quali possono arrivare ad una cooperazione. Concetto che analizzerò in seguito con il celebre dilemma del prigioniero. Un'importante applicazione di questa è che i giocatori sono indotti ad operare in generale con interessi non coincidenti. Avendo considerato il conflitto di interessi come la caratteristica fondamentale del gioco non cooperativo è futile rilevare che nel caso in cui tale contesto di interessi sia massimo, ovvero gli interessi sono esattamente contrapposti, si parla in codesto caso di gioco strettamente non cooperativo o definibile anche strettamente competitivo. Chiaramente questa situazione può verificarsi solo nel caso limite di due soli players, dunque in una situazione di duopolio. La teoria dei giochi è quella branca dell'economia applicata che studia il comportamento economicamente razionale dei players che si confrontano in modo continuo nel mondo economico, nel quale esistono numerosissimi vincoli e limiti che sostanzialmente condizionano in modo marcato le scelte alle quali sono chiamati gli stessi giocatori. Lo Scopo della teoria dei giochi è quello di trovare soluzioni a queste forme di conflitto di interessi contrapposti esistenti sui vari mercati. La teoria dei giochi parte dal presupposto che lo scopo dei players è vincere e prevede la conoscenza delle regole di un determinato gioco e dei risultati che possiamo ottenere in ogni singola situazione, i cosiddetti payoff. Ogni giocatore può decidere un numero limitato di azioni e questo determina la sua strategia, che può portare a diversi payoff che possono essere positivi in caso di guadagno o negativi in caso di perdita. Le diverse strategie possono essere rappresentate mediante una matrice, la cosiddetta matrice di "Payoff" in cui vengono inseriti i payoff corrispondenti a tutti i risultati possibili oppure mediante la ramificazione delle decisioni, in cui vengono rappresentate in modo sequenziale le decisioni con tutte le possibili combinazioni delle giocate. I giochi possono essere classificati in vari modi, ma in economia la classificazione più rilevante è quella che distingue tra cooperativi e non cooperativi. Un gioco si dice cooperativo se c'è la possibilità per i giocatori di sottoscrivere accordi vincolanti, che possono essere di vantaggio ai singoli giocatori. I giochi cooperativi si hanno dunque quando gli interessi dei giocatori non sono in contrapposizione diretta tra loro, ma esiste un interesse comune. Un gioco si dice invece non cooperativo quando il meccanismo delle decisioni riguarda i singoli giocatori sulla base di ragionamenti individuali. Ovvero le parti non possono stipulare accordi e sono in concorrenza tra di loro. In questa situazione ogni giocatore ha lo scopo di perseguire la strategia che risulta essere più vantaggiosa per lui; nel caso in cui esista una strategia che massimizzi i guadagni per tutti i giocatori

allora si parla di punto di equilibrio o “Equilibrio di Nash”. Nessun giocatore dotato di razionalità ha interesse a cambiare la strategia dominante.

Tuttavia, si deve sottolineare come l’equilibrio di Nash non sia il miglior riscontro per tutti: infatti se i players decidessero di allontanarsi congiuntamente dal punto di equilibrio potrebbero migliorare il proprio guadagno. Per spiegare la teoria dell’Equilibrio di Nash è bene citare una tesi di Adam Smith che secondo Nash era sbagliata, ovvero:” Il migliore risultato si ottiene quando il miglior componente del gruppo fa il meglio per sé” ma secondo Nash questa affermazione era incompleta in quanto secondo lui: “Ogni componente del gruppo otterrà il miglior risultato se farà il meglio per sé e per il gruppo”. Nel celebre film a Beautiful Mind, che narra della storia di Nash, il protagonista (John Nash) mentre si trova al bar con dei colleghi riesce a fare una scoperta sensazionale.

Nash e i tre suoi colleghi sono comodamente seduti al tavolo quando entrano quattro ragazze e tra di esse una spicca sulle altre in qualità di bellezza. Secondo i suoi colleghi l’ottimo paretiano sta nel provarci con la ragazza più bella e riuscire ad uscirci. L’ottimo di paretiano è una situazione di allocazione efficiente delle risorse. In un ottimo paretiano, non è possibile migliorare l’utilità di un soggetto senza peggiorare quella degli altri soggetti. L’ottimo di Pareto è razionale dal punto di vista collettivo, ma non lo è dal punto di vista individuale: nei giochi non cooperativi, in cui i soggetti agiscono col fine di massimizzare la propria utilità, anche a discapito degli altri, non sempre si raggiunge un ottimo di Pareto. Tornando al nostro esempio il punto di vista di Nash era totalmente differente, secondo lui l’ottimo paretiano per il gruppo in quella situazione sarebbe stata per ognuno uscire con una ragazza, senza far caso alla bellezza. Così facendo sia le ragazze che i ragazzi saranno soddisfatti e qualcuno addirittura potrebbe riuscire nell’impresa di uscire con la ragazza più carina. Se invece essi ci provassero tutti con la ragazza più carina, lei rimarrebbe infastidita e probabilmente nessuno raggiungerebbe l’ottimo di Pareto e allo stesso tempo le altre ragazze rimarrebbero scontente nel sentirsi la seconda scelta. Questo esempio ci aiuta a capire meglio come funziona l’Equilibrio di Nash nei giochi cooperativi, per applicarlo invece nei giochi non cooperativi possiamo utilizzare il celebre “Dilemma del prigioniero” che analizzeremo in seguito.

Cercheremo oggi di analizzare l’importanza della connessione tra la Teoria dei giochi e l’intelligenza artificiale (IA): ovvero la tecnologia di base che consente di simulare i processi dell’intelligenza umana attraverso la creazione e l’applicazione di algoritmi integrati in un ambiente di calcolo dinamico. In altre parole, l’obiettivo dell’AI è quello di creare computer in grado di pensare e agire come gli esseri umani. Le applicazioni della teoria dei giochi nell’IA servono per:

\* **Ottimizzazione delle Decisioni in Ambiti Complessi:** La teoria dei giochi consente di modellare e risolvere problemi di decisione strategica, che sono fondamentali per lo sviluppo di sistemi di IA

capaci di operare in ambienti competitivi e cooperativi. Questo è particolarmente rilevante in settori come l'economia, la finanza, la logistica e la gestione delle risorse, dove le decisioni ottimali dipendono dalle azioni di molteplici attori interconnessi.

\* **Sviluppo di Sistemi Multi-Agente:** Nei sistemi di IA multi-agente, la teoria dei giochi aiuta a progettare algoritmi che coordinano il comportamento di più agenti autonomi, garantendo che lavorino insieme in modo efficiente e raggiungano obiettivi comuni. Questo è essenziale in applicazioni come la robotica collaborativa, i sistemi di trasporto intelligente e la gestione delle reti energetiche.

\* **Apprendimento e Adattamento Dinamico:** La teoria dei giochi fornisce principi fondamentali per l'apprendimento per rinforzo e l'apprendimento imitativo, tecniche chiave nell'IA. Gli agenti di IA possono utilizzare modelli di gioco per migliorare continuamente le proprie strategie attraverso l'interazione con altri agenti e l'ambiente, imparando da esperienze passate e adattandosi a nuovi scenari.

## **Giochi cooperativi nell'Intelligenza Artificiale**

I giochi cooperativi non nascono per mettere in competizione i diversi soggetti, ma cercano di creare sinergie di gruppo, stimolando i giocatori verso la scoperta della collaborazione reciproca. Si evidenzia in questo gioco la possibilità di creare alleanze ed accordi per perseguire obiettivi comuni, ma affinché questo si verifichi i giocatori devono cooperare. Un gioco cooperativo mette in risalto tutte le possibilità che i player hanno nell'interagire fra loro, tenendo in considerazione che tutte le coalizioni sono ammissibili. Alcune volte i giocatori sacrificano il loro podio per puntare ad ottenere una vittoria di squadra; infatti, il gioco cooperativo rappresenta o la vittoria o la perdita comune, ma talvolta è possibile incontrare delle eccezioni. Quindi il gioco cooperativo è un tipo di gioco in cui i giocatori operano insieme per raggiungere un obiettivo comune anziché competere tra loro. Questo obiettivo comune viene raggiunto tramite la collaborazione, la comunicazione e la strategia, promuovendo il senso di squadra, la fiducia reciproca e la coesione di gruppo. L'intelligenza artificiale sta rivoluzionando diversi scenari, dall'assistenza sanitaria alla logistica, dall'industria automobilistica alla finanza. Uno degli scenari più interessanti in cui l'IA sta dimostrando il suo potenziale è relativo all'ambito dei giochi cooperativi. Tali giochi producono un campo abbastanza fertile per l'applicazione e l'attuazione degli algoritmi di IA, che devono coordinarsi e collaborare con molti fattori o giocatori, per creare obiettivi comuni. I giochi cooperativi mettono in luce il raggiungimento di alcuni traguardi, come: lo sviluppo di algoritmi di apprendimento collaborativo, la nascita di strategie di coordinamento e comunicazione tra gli agenti intelligenti. Nonostante i numerosi benefici, l'introduzione e la concretizzazione dell'IA nei giochi cooperativi definisce diverse sfide. Una di esse

è data dalla complessità della coordinazione tra agenti autonomi, che richiede la definizione di algoritmi in grado di negoziare, organizzare e allinearsi in modo dinamico alle azioni degli altri agenti. Un'altra sfida da dover affrontare è la necessità di sviluppare e definire meccanismi di fiducia e di risoluzione dei conflitti e contrasti tra questi agenti. Il gioco cooperativo nell'intelligenza artificiale è fondamentale per diversi motivi:

-Sviluppo di Algoritmi Collaborativi: I giochi cooperativi per fornire un eccellente sviluppo e progresso, si basano sull'osservazione di un ambiente controllato, mentre il test di algoritmi di IA è in grado di coordinarsi con altri agenti o sistemi. Questo è essenziale per la creazione di tecnologie più avanzate e sofisticate, che possono interagire e lavorare in sintonia con gli esseri umani e altri agenti intelligenti.

-Promozione della Ricerca Avanzata: La possibilità che hanno i diversi agenti nell'interagire nei giochi stimola la ricerca avanzata, in settori come la coordinazione multi-agente, la pianificazione collaborativa e la comunicazione distribuita. Questo crea progressi innovativi nella conoscenza e nell'applicazione effettiva e pratica dell'IA collaborativa.

-Sfida delle Capacità di Apprendimento: I giochi cooperativi danno un'ampia scelta di scenari e mansioni che richiedono un grande insieme di abilità cognitive. Tutto ciò si idealizza nella ricerca di una sfida stimolante per gli algoritmi di IA, portandoli a migliorare le loro capacità di apprendimento, adattamento e risoluzione dei problemi.

-Applicazioni Pratiche in Diversi Settori: Gli algoritmi di IA sono stati creati attraverso i giochi cooperativi, trovando applicazioni pratiche e concrete in un'ampia gamma di settori, tra cui la robotica collaborativa, la logistica, l'assistenza sanitaria e l'industria del gioco. Questi algoritmi offrono numerosi vantaggi tra cui: migliorare l'efficienza operativa, ottimizzare le risorse e migliorare l'esperienza degli utenti.

-Coltivazione di Competenze Sociali e di Teamwork: L'interazione che avviene in tali giochi promuove lo sviluppo e la crescita di competenze sociali e di teamwork, sia per gli agenti di IA che per gli esseri umani coinvolti. Questo è importante non solo per rafforzare le performance del gioco, ma anche per facilitare una maggiore integrazione e collaborazione tra uomini e macchine nella vita reale.

Dunque,  
se da un lato possiamo valutare i vantaggi che il gioco cooperativo apporta all'IA come:

-Collaborazione Dinamica: I giochi cooperativi forniscono un ambiente ben controllato in cui gli algoritmi di IA possono imparare a cooperare in tempo reale, accettando sfide complesse e adattandosi alle strategie degli altri agenti.

-Apprendimento Collaborativo: I giochi cooperativi definiscono lo sviluppo di algoritmi di IA in grado di apprendere, migliorare e rafforzarsi attraverso l'interazione di agenti, in maniera tale che questi ultimi si possano adattare ad una vasta gamma di contesti e scenari.

-Progettazione di Strategie: La natura collaborativa dei giochi richiede agli algoritmi di IA di sviluppare strategie efficaci di coordinamento e comunicazione, stimolando l'innovazione e la creatività nell'approccio alla risoluzione dei problemi;

dall'altro lato possiamo analizzare le sfide nell'implementazione di questi giochi nell'IA come:

-Complessità della Coordinazione: È complesso coordinare le azioni dei diversi agenti autonomi, in quanto quest'ultimo aspetto richiederebbe la realizzazione di algoritmi molto più sofisticati in grado di negoziare, pianificare e adattarsi alle situazioni dinamiche mutevoli.

-Comunicazione Effettiva: Gli algoritmi di IA puntano ad una comunicazione efficiente, trasmettendo informazioni cruciali e coordinando le azioni senza conflitti.

-Fiducia e Coesione del Team: La realizzazione di meccanismi di fiducia e solidarietà tra gli agenti di IA è fondamentale per garantire una collaborazione efficace e il perseguimento efficiente degli obiettivi comuni.

Diamo voce ad alcune applicazioni pratiche in cui l'intelligenza artificiale ha concretizzato le sue capacità all'interno dei giochi cooperativi. Nei giochi da tavola, nasce con l'obiettivo di creare un avversario in grado di simulare un comportamento razionale ed intelligente. Le prime applicazioni di intelligenza artificiale vengono presentate negli anni 50, quando Christopher Strachey crea un programma per il gioco della dama, mentre Dietrich Prinz ne presenta uno per gli scacchi. L'algoritmo di Prinz è in grado di trovare la soluzione migliore a due mosse dallo scacco matto. Successivamente Arthur Samuel studia un algoritmo di intelligenza artificiale per la dama, in grado di competere con giocatori principianti. Samuel introduce e studia un algoritmo di ricerca chiamato Minimax, un metodo per minimizzare la massima perdita possibile (o per massimizzare il minimo guadagno). Questo algoritmo costruisce un albero completo degli stati del gioco e analizza la decisione migliore facendo un calcolo ricorsivo. L'algoritmo assume che, dei due giocatori, il primo cerchi di massimizzare le proprie probabilità di vincita e viceversa, il secondo cerchi di minimizzarle. Il Minimax risulta essere molto dispendioso, soprattutto in domini in cui lo spazio degli stati è ampio, dato che l'albero deve essere completamente espanso e visitato. Per limitare il numero di nodi esaminati, Samuel utilizza la "potatura alfa-beta", che consiste nel terminare la valutazione di una possibile mossa appena viene dimostrato che è peggiore di una esaminata in precedenza. Samuel, volendo migliorare il proprio programma, introduce il metodo da lui chiamato "apprendimento

meccanico". Questa tecnica consiste nell'addestrare l'intelligenza artificiale facendola giocare migliaia di volte contro sé stessa. Negli anni 80 Jonathan Schaeffer et al iniziano a lavorare su Chinook, un programma per la dama, sviluppato per personal computer. Basato sulla ricerca alfa-beta e su un database pre-calcolato con più di 400 miliardi di posizioni con 8 pezzi in gioco o meno, il programma di Schaeffer diventa campione del mondo nel '94. Lo stesso gruppo di ricerca riesce a risolvere nel 2007 il gioco della dama (con scacchiera classica 8 x 8), dimostrando che la partita giocata senza errori nasce in parità. Gli scacchi sono uno dei giochi più usati ma risulta essere anche molto difficile. Il primo computer in grado di vincere un umano a scacchi è quello di Deep Thought nel 1989. L'algoritmo è capace di calcolare 100 milioni di posizioni al secondo. Le sue funzioni di valutazione sono state scritte con parametri determinati dal sistema stesso, analizzando migliaia di partite di campioni. Le prestazioni dei software di scacchi sono in continuo miglioramento. Nel 2009 il software Pocket Fritz 4, installato su uno smartphone, vince un torneo di categoria 6, riuscendo a valutare circa 20000 posizioni al secondo.

L'IA affina le sue doti anche nel campo della robotica.

I

I primi robot sono nati per aiutare l'uomo in fabbrica, sono stati impiegati per fare tutte quelle operazioni ripetitive o per aiutare un operatore umano di fronte a carichi troppo pesanti. L'utilizzo dei robot è sempre stato idealizzato nel settore industriale o nel settore automobilistico. Ricordiamo sempre che dietro l'utilizzo di una macchina robotica c'è un'elevata pericolosità, visto che si tratta sempre di macchine in movimento. Le prestazioni dei robot sono notevoli e crescenti, garantiscono un'alta precisione nelle diverse mansioni, con un tempo naturalmente ridotto. È possibile trovare macchine automatiche anche in ambienti tradizionali, la robotica, al giorno d'oggi si è completamente evoluta, basti pensare alla nascita dei robot per le pulizie domestiche. Grazie agli algoritmi dell'intelligenza artificiale è possibile sviluppare un sistema di elaborazione di immagini o video, in modo tale da estrarre informazioni, riconoscere oggetti e intraprendere azioni a partire dai segnali ricavati. L'obiettivo è quello di imitare la vita umana, ma a differenza di un essere umano, un sistema elettronico non dispone di tutta l'esperienza rafforzata nel corso della vita. Uno stesso oggetto può essere analizzato sotto diversi tipi di angolazioni, può essere in movimento o fermo e il suo colore può essere molto variabile. Questi parametri per noi sono abbastanza semplici da generalizzare, grazie all'esperienza umana. L'intelligenza artificiale, con i suoi sistemi e algoritmi, deve essere addestrata in un tempo molto breve, sono necessari infatti tantissimi dati e strumenti che il software deve studiare ed elaborare, come una serie di immagini da analizzare nelle quali l'oggetto si trova in configurazioni molto varie tra loro.

I giochi cooperativi e i giochi non cooperativi nell'intelligenza artificiale (IA) differiscono significativamente nella loro natura, obiettivi e dinamiche di gioco. Di seguito, mettiamo a confronto questi due approcci:

-Natura del

Gioco: nei giochi cooperativi, i partecipanti lavorano insieme per raggiungere un obiettivo comune. L'accento è sulla collaborazione, la condivisione di risorse e l'aiuto reciproco per superare sfide o completare un compito. Nei giochi non cooperativi, i partecipanti competono tra loro per raggiungere un obiettivo personale. L'obiettivo principale è battere gli altri giocatori, spesso attraverso strategie di competizione e sfruttamento delle debolezze degli avversari.

-Obiettivi del Gioco:

l'obiettivo principale dei giochi cooperativi è raggiungere un obiettivo comune come squadra. Il successo è misurato dalla capacità di collaborare efficacemente e di superare le sfide insieme. Nei giochi non cooperativi, l'obiettivo è principalmente vincere rispetto agli altri giocatori. Il successo è spesso misurato in base alla classifica o al punteggio personale, con poca o nessuna considerazione per il benessere degli altri giocatori.

-Dinamiche di gioco:

favoriscono la comunicazione, la pianificazione strategica e la condivisione di informazioni tra i partecipanti. Gli utenti devono imparare a fidarsi l'uno dell'altro e a coordinare le azioni per raggiungere l'obiettivo comune. Le dinamiche di gioco non cooperativo possono essere più competitive e individualistiche. Gli utenti tendono a proteggere le proprie risorse, a pianificare tattiche per superare gli avversari e a sfruttare le vulnerabilità degli altri per ottenere vantaggi. Applicazioni Pratiche: le applicazioni pratiche del gioco cooperativo nell'IA possono includere robotica collaborativa, sistemi di trasporto intelligente e simulazioni di squadra per situazioni di emergenza. Le applicazioni pratiche del gioco non cooperativo nell'IA possono includere algoritmi di intelligenza artificiale per la progettazione di strategie competitive, ad esempio nei giochi da tavolo o nei videogiochi competitivi online. In sintesi, mentre i giochi cooperativi si concentrano sulla collaborazione e sull'obiettivo comune, i giochi non cooperativi si concentrano sulla competizione e sul raggiungimento del successo individuale. Entrambi gli approcci hanno applicazioni significative nell'ambito dell'IA e offrono sfide uniche per lo sviluppo di algoritmi intelligenti. In entrambi i casi, l'IA può essere addestrata per affrontare sia situazioni cooperative che non cooperative, e la sua efficacia dipende spesso dalla capacità di adattarsi dinamicamente alle condizioni di gioco e alle strategie degli altri agenti.

## **Equilibrio di Nash**



Tra gli studiosi che hanno contribuito alla teoria dei giochi, fondamentale è senza dubbio l'apporto del matematico ed economista John Nash (1928) che ha introdotto per la prima volta nella sua tesi di laurea (1950) il concetto di Equilibrio di Nash.

L'equilibrio di Nash consiste in un insieme di soluzioni per i giochi non cooperativi tra due o più soggetti, in cui ciascun giocatore gioca in modo razionale e nessuno ha un interesse a cambiare la propria strategia. Se ogni giocatore sceglie una strategia e nessun altro ha un vantaggio nel cambiarla mentre tutti gli altri mantengono la propria, allora queste strategie ed i corrispettivi payoff costituiscono l'equilibrio di Nash. Dunque, Nash stabilisce che ciascun soggetto deve fare la scelta migliore per sé stesso tenendo conto di quelle di tutti gli altri; questo porta all'equilibrio.

Definiamo un gioco  $(S, f)$  come una coppia formata da  $S$ , ossia la combinazione strategica di ciascun giocatore, e da  $f$ , cioè la funzione di payoff (o di pagamento).

- La combinazione strategica  $S$  (o insieme di profili strategici) è data dall'insieme delle strategie di ciascun giocatore che definiscono tutte le decisioni che prenderà durante il gioco. Un profilo strategico comprende una sola strategia per ciascun giocatore, dove per strategia s'intende l'insieme di tutte le azioni che possono essere eseguite dal giocatore in tutte le situazioni possibili del gioco;
- La funzione di payoff  $f(x) = (f_1(x), f_2(x), \dots, f_n(x))$ , è la funzione che associa a ciascun giocatore  $i$  il payoff derivante da una data combinazione di strategie. Il payoff di ogni giocatore dipende, oltre che dalla propria strategia, anche da quelle adottate dai giocatori avversari.

Dunque, a questo punto:

Sia  $(S, f)$  un gioco con  $N = \{1, 2, \dots, n\}$  giocatori, dove  $S_i$  sia la strategia del giocatore  $i$ ,  $S = S_1 \times S_2 \times \dots \times S_n$  sia l'insieme delle combinazioni strategiche e  $f(x) = (f_1(x), f_2(x), \dots, f_n(x))$  sia la funzione dei payoff con  $x \in S$ , ossia l'insieme di tutte le strategie. Sia  $x_i$  il profilo strategico scelto dal giocatore  $i$  e  $x_{-i}$  il profilo strategico di tutti gli altri giocatori. Quando ciascun giocatore  $i \in 1, 2, \dots, n$  sceglie la strategia  $x_i$  all'interno del profilo delle strategie  $x = (x_1, x_2, \dots, x_n)$  allora ottiene un payoff  $f_i(x)$ . Dunque, il payoff di ogni giocatore dipende dal profilo strategico scelto.

Un profilo di strategia  $x^*$  è un equilibrio di Nash se ciascun giocatore cambiando singolarmente strategia non ne trae alcun vantaggio; ossia tale che, per ogni  $i \in N$  e per ogni  $x_i \in S_i$ , si abbia che:

$$f_i(x_i^*, x_{-i}^*) \geq f_i(x_i, x_{-i}^*)$$

L'applicazione dell'equilibrio di Nash che si riporta più comunemente in esempio è il dilemma del prigioniero. Rientra tra i giochi ad informazione completa, ossia quelli in cui tutti giocatori conoscono le regole del gioco, le strategie da adottare ed i payoff attesi, sia propri che quelli degli altri. Il dilemma del prigioniero è un gioco in cui vi sono due giocatori che vengono arrestati dalla polizia perché accusati di aver commesso un grave reato; i due sospettati sono detenuti in celle separate dunque non hanno la possibilità di comunicare tra di loro; ognuno può scegliere di confessare oppure non confessare. La scelta di ciascuno dei due giocatori influenza anche l'altro. Viene spiegato loro che:

- se solo uno dei due confessa, accusando l'altro di aver commesso il reato, il primo verrà liberato mentre il secondo sarà condannato a 10 anni di carcere;
- se nessuno dei due confessa, verranno accusati del reato minore e saranno condannati entrambi ad un anno;
- se entrambi confessano, verranno condannati entrambi a 5 anni di carcere.

|              | Confessa | Non confessa |
|--------------|----------|--------------|
| Confessa     | (5;5)    | (0;10)       |
| Non confessa | (10;0)   | (1;1)        |

Matrice di payoff del dilemma del prigioniero

Se il prigioniero confessa, quindi, rischia 0 o 5 anni di carcere, se non confessa rischia 1 o 10 anni. Ogni prigioniero, quindi, migliora la sua situazione confessando, sapendo che la migliore decisione per l'altro sia confessare; per Nash questo gioco ha un singolo equilibrio che si ottiene quando entrambi si tradiscono ossia quando entrambi confessano. Questa potrebbe sembrare una scelta poco razionale in quanto il risultato migliore per entrambi sarebbe quello di non confessare con un anno in prigione anziché cinque, ma questo non è un equilibrio. Quest'ultimo viene definito ottimo paretiano e si verifica quando presa una qualunque delle decisioni, non è possibile trovare un'altra soluzione che comporti per almeno uno dei due sospettati una riduzione degli anni di carcere senza che aumentino quelli dell'altro.

Per capire se in una data partita si è raggiunto l'equilibrio di Nash, una tecnica semplice, è quella di rivelare agli avversari la strategia adottata; se, nonostante ciò, nessuno di loro cambia strategia, l'equilibrio di Nash è dimostrato.

L'equilibrio di Nash è un aspetto della teoria dei giochi comunemente applicato in diversi contesti quali quello economico, informatico, medico, giuridico, sociale e così via; oggigiorno è spesso utilizzato, in una pluralità di modi, nell'ambito dell'Intelligenza Artificiale.

È possibile, infatti, usare degli algoritmi di apprendimento automatico per individuare l'equilibrio di Nash nell'addestramento degli agenti intelligenti in modo tale da saper rispondere in modo ottimale alle diverse situazioni. Oggi è inoltre molto frequente l'applicazione e l'utilizzo di algoritmi di Intelligenza Artificiale in casi pratici come gli scambi commerciali, per l'individuazione dell'equilibrio di Nash in modo da massimizzare il benessere sociale tramite l'ottimizzazione dell'allocazione delle risorse, la riduzione dei costi e dei tempi; oppure nei sistemi di protezione informatica, per individuare le strategie che consentano di raggiungere un equilibrio ossia quella situazione in cui l'hacker non abbia alcun vantaggio nel violare il sistema e dunque quest'ultimo sia il più protetto possibile.

Una limitazione dell'applicazione dell'equilibrio di Nash all'intelligenza artificiale è sicuramente il fatto che i giochi asimmetrici (in cui i giocatori hanno obiettivi diversi o contrastanti) sono più comuni nel mondo reale e di conseguenza nell'AI; tuttavia, è più facile da ottenere nei giochi simmetrici (in cui tutti i giocatori hanno gli stessi obiettivi e solo le loro strategie messe in atto per raggiungerli determineranno chi vince la partita).

Nonostante questo, l'equilibrio di Nash, come abbiamo detto, viene utilizzato per individuare le strategie ottimali per l'Intelligenza Artificiale in una serie di contesti.

Per ottenere queste strategie è necessario seguire un vero e proprio procedimento:

1. In primo luogo, è necessario individuare il gioco o l'ambiente di riferimento, con la conseguente conoscenza di tutti i dati, di tutte le regole e di tutte le informazioni necessarie per affrontarlo al meglio;
2. In secondo luogo, si devono individuare i giocatori e le possibili strategie per ciascuno di essi;
3. In terzo luogo, è importante determinare i payoff di ciascun giocatore.

A questo punto si deve applicare la teoria dei giochi; un algoritmo risolutivo assocerà un payoff ad ogni possibile strategia; perciò, dalla matrice dei payoff l'algoritmo sceglierà la strategia relativa all'equilibrio di Nash nel gioco considerato. Successivamente, occorre mettere in pratica il processo

che consente all'AI di adottare ed applicare la strategia ottimale individuata (tramite strumenti di AI); è un processo complesso che va dalla progettazione ed addestramento del modello fino alla sua integrazione nell'ambiente operativo. Per finire occorre valutare le prestazioni dell'intelligenza artificiale perché, se i risultati raggiunti non erano quelli sperati sarà necessario migliorare ed aggiornare il modello dell'AI in relazione ai comportamenti osservati.

Ed è così che, grazie all'equilibrio di Nash, l'intelligenza artificiale può trovare le strategie ottimali in una pluralità di ambienti, in modo da migliorare passo dopo passo le sue prestazioni e la sua integrazione; con lo sviluppo di algoritmi di AI in grado di massimizzare l'efficienza del contesto preso in considerazione.

### **Teoria dei giochi inversa**

La teoria dei giochi mira a comprendere le dinamiche di un gioco per ottimizzare il possibile risultato dei suoi giocatori. L'inverse game theory, la cosiddetta teoria dei giochi "inversa", mira invece a progettare un gioco basato sulle strategie e gli obiettivi dei giocatori. La teoria dei giochi inversi svolge un ruolo importante nella progettazione di ambienti degli agenti di intelligenza artificiale.

Uno degli obiettivi principali della teoria dei giochi è prevedere il comportamento degli agenti attraverso concetti di equilibrio in un dato gioco. Tuttavia, possiamo osservare alcuni comportamenti di equilibrio degli agenti, ma non è possibile comprendere le dinamiche che si instaurano quando un giocatore decide di compiere una scelta. Ad esempio, una piattaforma di shopping online può osservare le decisioni di acquisto degli acquirenti su diversi prezzi di vendita, ma la piattaforma ha una conoscenza limitata delle esatte utilità degli acquirenti. Allo stesso modo, mentre il policymaker può osservare le reazioni del mercato al suo annuncio, i motivi esatti dietro le reazioni dei trader sono solitamente poco chiari. Pertanto, recuperare i parametri di gioco sottostanti non solo ci porterebbe a decisioni strategiche migliori, ma migliorerebbe anche le nostre spiegazioni dei motivi delle scelte dei consumatori.

Queste prospettive motivano una classe di problemi di ricerca noti come teoria dei giochi inversi: dati i comportamenti di equilibrio degli agenti, quali sono le possibili utilità che inducono questi stessi comportamenti? Ci concentriamo specificamente sull'impostazione del gioco sequenziale dal punto di vista dell'agente che si muove per primo (ad esempio, piattaforma Internet, policymaker o difensore) le cui diverse strategie (ad esempio, prezzo, regolamentazione o schema di difesa) indurrebbero diversi comportamenti di equilibrio. del seguente agente (ad esempio, utenti Internet, commercianti o aggressori). Gli studi su tali impostazioni di gioco hanno visto ampi impatti e applicazioni estese che vanno dai problemi principale-agente nella progettazione dei contratti, l'AI Economist ai giochi di sicurezza modellati per il bene sociale.

Formalizziamo il nostro problema nella forma normale del gioco Stackelberg, in cui un leader ha il potere di impegno di una strategia randomizzata e un seguace decide di conseguenza la sua risposta. È noto che l'impegno ottimale del leader può essere calcolato in modo efficiente in un singolo programma lineare, data la piena conoscenza del gioco. Tuttavia, il problema dell'apprendimento inverso per determinare il gioco sottostante dalle risposte del follower è più impegnativo: Letchford et al., Peng et al. mostrano che l'apprendimento della strategia ottimale del leader dalle migliori risposte del follower richiede un numero di campioni che rappresenta un polinomio di alto grado nella dimensione del gioco e può essere esponenziale nei casi peggiori. Ciò limita in modo significativo la praticità di questi algoritmi, poiché il leader di solito non può permettersi il tempo o i costi per raccogliere feedback da così tante interazioni.

Più preoccupante è la scomoda realtà che difficilmente possiamo aspettarci risposte di equilibrio ottimali degli agenti ipotizzate nel lavoro esistente. In effetti, questi stessi acquirenti, commercianti o aggressori difficilmente conoscono le loro esatte utilità e sono naturalmente incapaci di determinare la strategia di massimizzazione dell'utilità attesa.

Tra i vari modelli per quantificare i comportamenti irrazionali degli agenti, uno dei più popolari è forse il modello di risposta quantistica (QR), che adotta il noto modello di scelta logit per catturare la selezione probabilistica delle azioni degli agenti. Il successo della modellazione del modello QR è attribuibile alle proprietà matematiche e statistiche della funzione logit che può catturare una varietà di comportamenti limitatamente razionali sotto diversi parametri. QRE è ampiamente adottato soprattutto nei giochi Stackelberg (di sicurezza), nei giochi a somma zero e in particolare è stato implementato in varie applicazioni del mondo reale. Inoltre, la struttura del modello QR è stata utilizzata anche in vari altri contesti, come l'attivazione softmax nella rete neurale, la regressione logistica multinomiale e l'algoritmo di aggiornamento del peso moltiplicativo per l'apprendimento senza rimpianti.

L'intuizione chiave è che lo strato di complessità comportamentale dovuto alla razionalità limitata, mentre complica la modellazione e il calcolo, fornisce una fonte più informativa per noi per apprendere l'utilità sottostante degli agenti. Per comprendere le intuizioni e le motivazioni dietro i nostri risultati, consideriamo il caso in cui il follower ha un'azione dominata  $j_1$ , dove l'utilità del profilo d'azione del leader e del follower  $(i, j)$  è specificata da  $u_{i,j}, v_{i,j}$  rispettivamente. Convenzionalmente, un'istanza di questo tipo viene trattata come un'istanza degenerata, perché il leader potrebbe ignorare l'azione  $j_1$  che un seguace perfettamente razionale non giocherebbe mai. Quindi, la strategia ottimale del leader è chiaramente quella di giocare sempre l'azione  $i_2$ . Tuttavia, di fronte ad un seguace limitatamente razionale, diventa possibile osservare la risposta  $j_1$  e stimare le utilità relative a questa azione dominata. Ad esempio, se il follower esegue le sue azioni  $j_1$  e  $j_2$  quasi

con la stessa frequenza, l'utilità attesa del follower sulle due azioni dovrebbe essere vicina. Sebbene tale azione dominata non abbia alcun effetto sulla strategia ottimale del leader contro un seguace perfettamente razionale, potrebbe essere un'azione potenzialmente dannosa (o benefica) che il leader voglia evitare (o incoraggiare) a svolgere un seguace razionale limitato. Cioè, nell'esempio di gioco sopra, se un seguace un po' irrazionale gioca l'azione  $j_1$ , sarebbe pericoloso per il leader giocare l'azione  $i_2$  ma sarebbe gratificante giocare l'azione  $i_1$ ; pertanto, una strategia leader più solida dovrebbe randomizzare assegnando una certa probabilità di svolgere l'azione  $i_1$ .

Notiamo che in generale, anche senza questo caso estremo di azioni dominate, sono ora disponibili informazioni extra sul payoff su quanto sia peggiore (o migliore) utilizzare la frequenza empirica delle risposte dell'azione limitatamente razionale, che vengono trascurati sotto il presupposto di seguaci perfettamente razionali.

A livello concettuale, possiamo notare che i rumori dovuti a comportamenti razionali limitati potrebbero essere sfruttati come fonti di informazione aggiuntive per apprendere l'utilità del follower. Il problema dell'apprendimento nei giochi sequenziali è stato studiato in diverse configurazioni. Marecki et al., Balcan et al. considerano il problema dell'apprendimento online nel gioco di sicurezza di Stackelberg con tipi di follower scelti in modo contraddittorio. Bai et al. considerano un ambiente di apprendimento bandito in cui si potrebbe interrogare qualsiasi voce dell'utilità dei follower sotto rumore e utilizzare la stima dell'utilità per approssimare la strategia ottimale del leader; tuttavia, questo processo di apprendimento presuppone la centralizzazione, cioè il learner può controllare sia le azioni del leader che quelle dei follower. Più simile alla nostra è l'impostazione dell'apprendimento strategico nei giochi di Stackelberg studiati, in cui il leader sceglie in modo adattivo le sue strategie in base all'osservazione della migliore risposta del follower e alla fine recupera l'utilità del follower fino a un certo livello di precisione.

Vorobeychik et al si sono occupati di trattare la tematica della teoria inversa dei giochi: hanno considerato il problema dell'apprendimento della funzione payoff utilizzando i profili strategici e le utilità corrispondenti. Kuleshov e Schrijvers introdussero il concetto di teoria dei giochi inversa, e gli autori dimostrarono che il problema di calcolare le utilità degli agenti da un insieme di equilibrio correlato è NP-Hard, a meno che il gioco non sia noto per avere strutture speciali. Più recentemente, il problema della teoria dei giochi inversa è stato studiato con il modello QR e ha portato ad alcuni risultati positivi: Sinha et al. considerano la configurazione di apprendimento PAC offline in cui le risposte dei follower possono essere previste con un piccolo errore per una distribuzione strategia di leader fissa. Haghtalab et al. dimostrano che solo tre strategie sono sufficienti per recuperare le funzioni di payoff dei follower lineari nei giochi di sicurezza; Ling et al. presentano un framework di apprendimento end-to-end che apprende il payoff del gioco a somma zero dal suo QRE. I problemi

inversi hanno ricevuto un'attenzione significativamente maggiore nei problemi decisionali con un singolo agente. Il problema più notevole è l'apprendimento per rinforzo inverso. Ci poniamo l'obiettivo primario di apprendimento come quello di recuperare una caratterizzazione completa dell'utilità del follower; i risultati spiegano come sia inutile e quasi irrealistico aspettarsi un esatto recupero dell'utilità del follower. Tale caratterizzazione dell'utilità può essere utilizzata per calcolare la strategia del leader ottimo sia sotto la razionalità perfetta, nota come equilibrio di Stackelberg forte (SSE), sia la razionalità limitata, nota come equilibrio quantico di Stackelberg (QSE).

Tale problema di apprendimento è stato considerato in particolare per i giochi di sicurezza di Stackelberg, dove il payoff è una funzione di utilità lineare unidimensionale strettamente semplificata.

## **Minmax**

MINAMX è una regola decisionale finalizzata a ridurre al minimo la possibile perdita massima, in un contesto di conflitto con agenti rivali che puntano a ottenere il massimo guadagno. Originariamente formulato per la teoria dei giochi a somma zero a più giocatori, coprendo sia i casi in cui i giocatori effettuano mosse alternative sia quelli in cui effettuano mosse simultanee, è stato esteso anche a giochi più complessi e al processo decisionale generale in presenza di incertezza. Trova applicazione in altri ambiti, come nel caso dell'intelligenza artificiale (di seguito IA). Nel campo dell'IA, si esaminano soprattutto giochi con le seguenti caratteristiche:

- informazione perfetta. Qui un giocatore conosce la propria posizione e quella dei suoi avversari. Degli altri giocatori, conosce anche lo storico delle mosse effettuate in passato. Quello che non conosce è le azioni che andranno a intraprendere in futuro. A questo, però, può ovviare facendo congetture sulla loro strategia di gioco, sulla base della quale potrà adeguare le proprie mosse.
- somma-zero. In un gioco con due avversari, se un giocatore vince l'altro ha perso inevitabilmente perso. Matematicamente, si potrebbe attribuire il valore 1 alla vittoria, 0 al pareggio e -1 alla sconfitta. Sulla base di questa assunzione, la somma dei valori dell'esito della partita sarà sempre pari a 0.
- sequenziale. I giocatori sono a conoscenza dell'ultima azione effettuata dall'altro;
- deterministico. Nel gioco non ci sono elementi casuali: il gioco dipende dalle scelte attuate dai giocatori i quali, avendo tutte le informazioni utili, possono determinarne anche le conseguenze
- non cooperativo. I giocatori sono indipendenti e non collaborano tra di loro, decidendo di massimizzare quindi il proprio pay-off e minimizzare quello degli altri.

Un esempio di gioco con tutte queste caratteristiche è quello degli scacchi, a cui può essere applicato l'algoritmo MINMAX. L'algoritmo consente di determinare le azioni migliori per i due giocatori partendo dall'analisi dei possibili modi in cui si può chiudere il gioco, fino a risalire alla posizione attuale dei giocatori. Questo avviene attraverso un'analisi ricorsiva dell'albero di gioco. L'albero di gioco è strutturato in nodi e rami. Negli scacchi, per esempio, ad ogni nodo corrisponde una diversa conformazione dei pezzi sulla tavola da gioco, mentre ad ogni ramo un'azione che fa passare i giocatori da un nodo all'altro. Il nodo che definisce lo stato iniziale prende il nome di nodo radice. Attraverso un processo di espansione del nodo, si aggiungono via via altri rami e nodi secondo un rapporto gerarchico tra gli stessi: dal nodo padre – generatore degli altri nodi - , infatti, nascono

ulteriori nodi chiamati nodi figli. Se non è più possibile espandere il nodo, oppure è stato semplicemente deciso di non generarne degli altri, il nodo prende il nome di nodo foglia. L'insieme dei nodi foglia prende il nome di frontiera.

Ipotizziamo di avere due giocatori: X e Y. Se tocca al giocatore X fare la sua mossa al livello 0 (nodo radice), sarà lui a creare gli altri nodi che andranno a costituire il livello 1 dell'albero di gioco, con l'obiettivo di massimizzare il valore dello stato. Al livello 1, toccherà al giocatore Y agire e creare i nodi che andranno a costituire il livello 2, con l'obiettivo in questo caso di minimizzare il valore dello stato (se fosse stato il giocatore Y a dare inizio al gioco, si sarebbe scelto il valore minore). Si prosegue finché non si raggiunge il livello di profondità prefissato. Arrivati alla frontiera, si attribuisce un valore arbitrario a ciascun nodo foglia. Dal momento che è stato il giocatore X a far partire il gioco, si sceglie lo stato con il valore maggiore e si porta al livello superiore  $n-1$ . Si continua così finché non si raggiunge la posizione iniziale  $n-n$  (livello 0). Sfruttando l'algoritmo MINMAX, questa sequenza indica il percorso migliore disponibile).

In merito al tema dell'esplorazione dell'albero, è importante capire quale nodo esplodere per primo. L'algoritmo di ricerca ci aiuta in questa scelta. Si distinguono gli algoritmi di ricerca che non hanno informazioni aggiuntive sulla soluzione (*algoritmi non informati*) e gli algoritmi che ne hanno (*algoritmi informati*). Tra i primi, che non sono visti come AI, ci sono:

- *depth-first* (prima-in-profondità). Si continua ad espandere il primo nodo generato, finché non si arriva a un nodo terminale (che non può essere più espanso) o a un nodo soluzione. A questo punto, possiamo passare ad espandere gli altri nodi generati dallo stesso padre e prova ad andare fino in profondità. Se tutti i nodi sono terminali non è stata identificata la soluzione, si riparte dal nodo superiore tralasciando ogni volta i nodi scartati. L'algoritmo non ha memoria i percorsi creati da ricerche fallimentari. Applicato insieme al MINMAX, è importante dare ai giocatori un limite massimo di profondità nell'esplorazione dell'albero e non fermarsi quando si trova un nodo soluzione, continuando a valutare tutti i nodi sulla frontiera.
- *breadth-first*. Si espandono tutti i nodi di un livello, prima di passare al successivo, finché non si trova un nodo soluzione. L'algoritmo memorizza tutti i nodi che ha esplorato. Prima di applicare MINMAX, è importante che l'algoritmo abbia creato l'intero albero fino alla profondità predefinita.
- L'Iterative deepening (approfondimento iterativo) unisce *depth-first* e *breadth-first*, consentendo di identificare la soluzione più vicina alla radice e quindi il percorso meno lungo grazie alle caratteristiche di *breadth-first* e di avere la complessità spaziale di *depth-first*.

Tra i secondi, troviamo i *best-first*, che hanno informazioni aggiuntive a quelle date dal problema utili per definire quali nodi espandere prima. L'obiettivo principale è identificare lo stato migliore (foglia principale) e procedere da qui con l'espansione. Quando si espande una foglia principale si ricalcola il valore MINMAX mettendo a confronto tutti i nodi sulla frontiera. Il nodo il cui valore raggiunge la radice diventa la nuova foglia principale ed è il prossimo nodo ad essere espanso.

Uno dei problemi che può riscontrarsi durante le ricerche con limite di profondità è il cosiddetto effetto orizzonte. Nel momento in cui si verifica un problema che non ha soluzione, continua a ritardare l'inevitabile perdita finché non viene superato il limite di ricerca prefissato, uscendo dalle situazioni analizzabili dall'algoritmo e facendo sembrare che il problema sia scomparso, quando in realtà è stato semplicemente spostato oltre l'orizzonte degli eventi visibili dall'algoritmo e molto probabilmente sarà riscontrato di nuovo nella successiva scansione. Un modo per migliorare la profondità dell'albero di ricerca è utilizzare estensioni singole, ovvero espandere ulteriormente mosse considerate indubbiamente migliori rispetto ad altre in una data posizione e memorizzarle una volta



scoperte. Quando si raggiunge il limite normale della ricerca, l'algoritmo controlla se la singola estensione è legale e, in caso affermativo, la considera. Anche se queste estensioni rendono l'albero più profondo, poiché sono poche, non vengono aggiunti molti nodi complessivi all'albero.

## **Sentiment Analysis**

La Sentiment Analysis (Analisi del sentiment) è un campo dell'Intelligenza Artificiale e del Natural Language Processing (Elaborazione del Linguaggio Naturale) che si occupa di identificare, estrarre e comprendere le emozioni, le opinioni e le tendenze soggettive espresse nei testi scritti. L'obiettivo principale della Sentiment Analysis è quello di determinare se il tono generale di un determinato testo è positivo, negativo o neutro. Questa tecnica viene comunemente utilizzata per analizzare opinioni, recensioni, commenti sui social media, articoli di notizie e altre forme di testo, al fine di comprendere il sentimento o l'atteggiamento del pubblico verso un particolare argomento, prodotto, servizio o evento. La Sentiment Analysis può essere realizzata mediante l'uso di algoritmi di machine learning, processi di linguistica computazionale e analisi statistica dei testi, che consentono di identificare e classificare le parole, le frasi o i documenti in base al loro contenuto emotivo.

Questo strumento ha numerose applicazioni pratiche, tra cui l'analisi del feedback dei clienti, il monitoraggio dell'umore, il supporto decisionale per il marketing e la valutazione del sentiment degli investitori nei confronti di determinate aziende o mercati finanziari. Quindi, la Sentiment Analysis fornisce un mezzo efficace per estrarre informazioni preziose dalle grandi quantità di dati testuali disponibili online e per comprendere meglio le opinioni e le percezioni delle persone.

Il processo di Sentiment Analysis è diviso in cinque fasi:

1. raccolta dati;
2. revisione dei testi;
3. individuazione del sentiment;
4. classificazione del sentiment;
5. presentazione dei risultati.

La fase di raccolta di dati consiste nell'estrarre dai social media come forum, blog, recensioni e dai social network tutte le opinioni che sono in qualche modo coerenti con l'oggetto di ricerca attraverso l'utilizzo di specifici software.

La seconda fase, ovvero quella di revisione dei testi, si esplica in una "pulizia" dai testi delle opinioni, eliminando innanzitutto le generalità degli opinion holder e tutte le informazioni non necessarie.

Nella fase successiva di individuazione del sentiment l'obiettivo è comprendere il sentiment delle varie opinioni per poterle in seguito classificare. L'individuazione può essere effettuata su più livelli:

1. livello del documento;

2. livello della frase;
3. livello della caratteristica.

La fase che segue riguarda la classificazione dei sentiment ottenuti.

Spesso si rappresentano con un'impostazione "positivo, negativo; buono, cattivo; like, dislike", ma esistono anche altri metodi, come per esempio il sistema di rating a stelle.

Infine, nell'ultima fase si analizza e si adatta il risultato per lo scopo preposto.

Le metodologie principali per analizzare un testo tramite la Sentiment Analysis sono:

- spotting di parole chiave, che classifica termini specifici come "divertito", "annoiato" oppure "gradevole" o "sgradevole";
- affinità lessicali, che assegnano alle parole rilevate un'emozione particolare;
- metodi statistici, con i quali da un'analisi sintattica profonda del testo si estrapolano opinioni in contesti più ampi;
- tecniche concettuali, che si basano sui tratti ontologici e semantici e sono in grado di comprendere sottigliezze espressive.

La Sentiment Analysis offre una serie di vantaggi significativi, ma presenta anche alcuni svantaggi che è importante considerare.

I vantaggi sono:

- la comprensione del sentiment del cliente, che consente alle aziende di comprendere meglio le opinioni dei propri clienti per adattare i loro prodotti, servizi e strategie di marketing;
- il monitoraggio dell'umore pubblico, attraverso l'analisi dei social media, dei forum online e degli articoli di notizie;
- la valutazione della reputazione online, con la quale le aziende possono valutare la propria reputazione online, monitorando le opinioni e le recensioni dei clienti sulle diverse piattaforme;
- l'automazione dei processi, che consente alle aziende di elaborare grandi quantità di dati testuali in modo efficiente, risparmiando tempo e risorse umane;
- il supporto decisionale, grazie al quale le aziende possono adattarsi rapidamente ai cambiamenti nelle opinioni del mercato tramite le informazioni estratte attraverso la Sentiment Analysis.

Invece, gli svantaggi sono:

- la complessità del linguaggio naturale, infatti, essendo il linguaggio umano complesso e ambiguo, può essere difficile per gli algoritmi comprendere il contesto e l'intento di un testo. Questo può portare a interpretazioni errate o a una classificazione non accurata del sentiment;
- il bilanciamento delle classi, in quanto, in alcuni casi, il numero di campioni positivi, negativi e neutri può essere sbilanciato, rendendo difficile l'addestramento di modelli accurati e bilanciati;
- la soggettività del sentiment, che può rendere difficile la creazione di un modello che sia universalmente accurato per tutti i tipi di testo e per tutti i contesti;
- la sensibilità alla lingua e alla cultura dei modelli, che può renderli meno precisi quando applicati a testi in lingue o contesti culturali diversi da quelli per cui sono stati addestrati.

In economia, la prima disciplina che mirò a comprendere il comportamento umano in contesti diversi, prima ancora della Sentiment Analysis, fu la Teoria dei Giochi. Entrambi sono ambiti distinti della scienza computazionale e sociale e integrare queste due discipline può offrire un'analisi più profonda e accurata delle dinamiche sociali. Ad esempio, in una piattaforma di social media in cui gli utenti possono interagire pubblicamente attraverso post e commenti, attraverso la Sentiment Analysis è possibile valutare il tono generale di una discussione e, applicando i principi della Teoria dei Giochi, si può modellare il comportamento degli utenti come decisioni strategiche, dove ciascun individuo cerca di massimizzare il proprio vantaggio sociale in base alle azioni degli altri partecipanti.

Le varie combinazioni di queste due discipline sono:

- la modellazione strategica per comprendere meglio il comportamento degli agenti in contesti sociali, politici o economici. Ad esempio, nel campo della politica, la Sentiment Analysis può essere utilizzata per valutare il supporto pubblico per determinate politiche o partiti politici per poi integrare queste informazioni nei modelli di Teoria dei Giochi per prevedere i risultati e ottimizzare le strategie;
- i giochi di opinione, infatti in alcuni contesti, le opinioni pubbliche e le emozioni possono essere viste come "giocatori" in un gioco in cui cercano di influenzare gli altri attori. Utilizzando la Sentiment Analysis, è possibile analizzare come le opinioni e le emozioni si diffondono e cambiano nel tempo in risposta a diverse azioni e stimoli e in seguito si può modellare questo processo, utilizzando concetti della Teoria dei Giochi come equilibrio Nash o strategie dominanti;
- la previsione di mercato in ambito finanziario, per la quale può essere utilizzata la Sentiment Analysis, analizzando il sentiment degli investitori e degli operatori di mercato e

l'integrazione di ciò in modelli di Teoria dei Giochi per prevedere le tendenze di mercato e ottimizzare le strategie di investimento.

Un'applicazione più specifica della Sentiment Analysis nella Teoria dei Giochi consiste nell'analisi del sentiment dei giocatori coinvolti nel Dilemma del prigioniero. Questo classico esempio coinvolge due prigionieri che devono decidere se cooperare o tradire il compagno di cella. Se entrambi cooperano, ricevono una sentenza ridotta; se entrambi tradiscono, ricevono una sentenza più severa; se uno tradisce e l'altro collabora, il traditore riceve un premio mentre il collaboratore una pena più pesante. L'analisi del sentiment potrebbe rivelare le emozioni e le aspettative dei prigionieri riguardo alla loro reciproca cooperazione o tradimento. Ad esempio, se un prigioniero nutre fiducia nel compagno e prevede che quest'ultimo si comporti in modo cooperativo, potrebbe essere incline a cooperare anche lui. D'altra parte, se si sente tradito o non fiducioso, potrebbe essere portato a tradire per proteggere i propri interessi. La Sentiment Analysis potrebbe identificare parole chiave nei loro comportamenti non verbali che suggeriscono sentimenti di fiducia, sfiducia, paura o vendetta.

In sintesi, l'applicazione della Sentiment Analysis nel Dilemma del Prigioniero offre un'opportunità per esplorare l'interazione tra razionalità strategica e sfere emotive nell'ambito della Teoria dei Giochi. Integrare la comprensione dei sentimenti dei giocatori potrebbe arricchire l'analisi delle dinamiche decisionali e delle implicazioni sociali di questo classico scenario.

Dunque, la combinazione della Sentiment Analysis con la Teoria dei Giochi offre una prospettiva più completa e sofisticata per comprendere e prevedere il comportamento umano in contesti complessi e dinamici. Questo approccio non solo migliora la comprensione delle interazioni sociali, ma offre anche opportunità pratiche per migliorare la gestione delle risorse e lo sviluppo di strategie più efficaci in vari settori.

### **Reti Generative Avversarie (GAN)**

Le reti avversarie generative sono un tipo di algoritmo di intelligenza artificiale progettato per risolvere il problema della modellazione generativa. L'obiettivo di un modello generativo è studiare una raccolta di esempi di addestramento e apprendere la distribuzione di probabilità che li ha generati. Le reti generative avversarie (GAN) sono quindi in grado di generare più esempi dalla distribuzione di probabilità stimata. Sebbene i modelli generativi basati sul deep learning siano comuni, le Gan sono tra i modelli generativi di maggior successo (soprattutto per la loro capacità di generare immagini realistiche ad alta risoluzione). Altri approcci di modellazione generativa si basano sull'ottimizzazione, mentre le Gan si basano sulla teoria dei giochi, che continua a presentare sfide e opportunità di ricerca uniche.

Le reti generative avversarie (Gans) sono una potente classe di modelli generativi introdotta da Goodfellow nel 2014. Il principio di base di Gans è ispirato al gioco a due giocatori a somma zero, in cui i guadagni totali di due giocatori sono pari a zero, e il guadagno o la perdita di utilità di ogni giocatore è esattamente bilanciato dalla perdita o guadagno dell'utilità di un altro giocatore.

I Gans sono costituiti da un generatore e da un discriminatore che apprendono simultaneamente. Il generatore cerca di catturare la distribuzione potenziale di campioni reali e genera nuovi campioni di dati. Il discriminatore è spesso un classificatore binario, che identifica i campioni reali dai campioni generati nel modo più accurato possibile. Sia il generatore che il discriminatore possono adottare la struttura di una rete neurale profonda. Il processo di ottimizzazione di Gans è un processo di gioco Minimax, in cui l'obiettivo dell'ottimizzazione è raggiungere l'equilibrio di Nash. È possibile utilizzare Gans per generare immagini di oggetti fotorealistici come uccelli e volti, generare scene interne o esterne, trasformare immagini dal dominio di origine al dominio di destinazione e generare immagini ad alta definizione da immagini a bassa definizione e così via.

-I modelli generativi svolgono un ruolo essenziale nel campo dell'IA, e i metodi generativi da soli sono di grande importanza per la ricerca. I metodi generativi e i metodi discriminativi sono due rami dell'apprendimento supervisionato nell'ambito dell'apprendimento automatico. I modelli generativi sono modelli ottenuti imparando con i metodi generativi. I metodi generativi coinvolgono ipotesi di distribuzione e stima dei parametri e possono campionare nuovi dati dai modelli stimati. Esistono due prospettive di ricerca: "gli esseri umani comprendono i dati" e "le macchine comprendono i dati." Dal punto di vista della "comprensione umana dei dati", un approccio tipico consiste nell'assumere una distribuzione utilizzando dati reali per adattare i parametri di distribuzione o addestrando il modello contenente la distribuzione. Dopo di che, un nuovo campione viene generato utilizzando la distribuzione appresa o modello. I metodi appartenenti a questa classe di modelli generativi includono la stima della massima probabilità, l'inferenza approssimativa e il metodo a catena di Markov. Nonostante i modelli appresi abbiano distribuzioni comprensibili all'uomo, l'apprendimento automatico ha i suoi limiti. Ad esempio, la stima della massima verosimiglianza viene eseguita su campioni di dati reali e i parametri vengono aggiornati direttamente in base ai campioni di dati, portando a un modello generativo eccessivamente liscio. Il modello generativo ottenuto con l'inferenza approssimativa possono solo avvicinarsi al limite inferiore della funzione obiettivo. Il metodo della catena di Markov può essere utilizzato per formare modelli generativi e generare nuovi campioni, ma la sua complessità computazionale è estremamente elevata. Dal punto di vista delle "macchine capiscono i dati", il modello generativo non stima o si adatta direttamente alla distribuzione dei dati. Al contrario, genera campioni di dati dalla distribuzione senza ipotesi esplicite e utilizza i campioni generati per modificare il modello. Si ipotizza che le macchine comprendano i dati in un modo che l'uomo non può capire esplicitamente, ma generare nuovi dati che l'uomo può capire. Prima di Gans, i modelli generativi costruiti dal punto di vista di "macchine capiscono i dati", dovevano essere addestrati utilizzando la catena di Markov, erano meno efficienti e avevano applicazioni sistematiche limitate. L'idea principale di GAN deriva dall'equilibrio di Nash nella teoria dei giochi. Assume due partecipanti al gioco: un generatore e un discriminatore. Il generatore ha lo scopo di apprendere la distribuzione dei dati reali, mentre il discriminatore mira a determinare correttamente se i dati di input sono dai dati reali o dal generatore. Per vincere il gioco, i due partecipanti devono continuamente ottimizzare sé stessi per migliorare la capacità di generazione e la capacità di discriminazione, rispettivamente. Lo scopo del processo di ottimizzazione è quello di

trovare un equilibrio di Nash tra i due partecipanti. Qualsiasi funzione differenziabile può essere utilizzata come generatore e discriminatore. Immaginiamo che le funzioni differenziabili  $D$  e  $G$  rappresentino il discriminatore e il generatore, e che i loro input siano rispettivamente i dati reali  $x$  e variabili casuali  $z$ , rispettivamente.  $G(z)$  rappresenta il campione generato da  $G$  e obbedisce ai dati di distribuzione dei dati reali. Se il discriminatore  $D$  proviene dai dati reali  $x$ ,  $D$  dovrebbe classificarlo come vero ed etichettarlo come  $1$ . Se l'input proviene da  $G(z)$ ,  $D$  lo classifica come falso e lo etichetta come  $0$ . Lo scopo di  $D$  è ottenere una classificazione certa della fonte dei dati, mentre lo scopo di  $G$  è quello di rendere le prestazioni dei dati generati  $G(z)$  su  $D$  (cioè  $D(G(z))$ ) coerenti con le prestazioni dei dati reali  $x$  su  $D$  (cioè,  $D(x)$ ). Il processo di ottimizzazione avversaria migliora gradualmente le prestazioni di  $D$  e  $G$ . Alla fine, se le prestazioni di identificazione di  $D$  è stata migliorano fino a un livello elevato ma non riescono a identificare correttamente la fonte di dati, si considera che il generatore  $G$  abbia catturato la distribuzione di dati reali.

-La scoperta di nuove applicazioni per l'addestramento avversario delle reti profonde è un'area di ricerca attiva. Queste applicazioni evidenziano diversi approcci all'uso delle rappresentazioni basate su GAN per la manipolazione, l'analisi o la caratterizzazione delle immagini e non riflettono completamente la gamma potenziale dell'applicazione delle Gan. L'utilizzo delle Gan per la classificazione delle immagini li colloca nel contesto più ampio dell'apprendimento automatico e fornisce un'utile valutazione quantitativa delle caratteristiche estratte nell'apprendimento non supervisionato. La sintesi delle immagini rimane una funzionalità importante delle Gan ed è particolarmente utile quando l'immagine generata può essere soggetta a vincoli preesistenti. Infine, la traduzione da immagine a immagine dimostra come i GAN offrano una soluzione generale a una serie di compiti che richiedono la conversione automatica di un'immagine di input in un'immagine di output.

#### 1) Classificazione e regressione.

Una volta completato l'addestramento di una Gan, la rete neurale può essere riutilizzata per altre attività a valle. Ad esempio, gli output degli strati convoluzionali del discriminatore possono essere utilizzati come estrattore di caratteristiche, con semplici modelli lineari sono adattati a queste caratteristiche utilizzando una ragionevole quantità di coppie (immagini, etichette). Utilizzando questo approccio, sono stati ottenuti buoni punteggi di classificazione su set di dati sia supervisionati che semi-supervisionati, anche quelli disgiunti dai dati di addestramento originali. Tuttavia, quando l'apprendimento avversario prevede l'apprendimento congiunto dei meccanismi di inferenza, come in ALI, la qualità della rappresentazione dei dati migliora; gli ultimi tre strati nascosti del codificatore ALI, simili al classificatore L2-SVM, sono stati utilizzati per costruire i vettori di rappresentazione, con un tasso di errori di classificazione significativamente ridotto. Inoltre, ALI ha ottenuto risultati di classificazione all'avanguardia per quanto riguarda le informazioni di etichettatura incorporate nella routine di allenamento. Quando i dati di addestramento etichettati sono in quantità limitata, è possibile utilizzare anche l'addestramento contraddittorio per sintetizzare più campioni di addestramento. Alcuni studiosi utilizzano i Gans per perfezionare le immagini sintetiche, mantenendo le informazioni sulle annotazioni. Addestrando i modelli solo su immagini sintetiche rifinite dal Gan (ovvero senza dati di addestramento reali), hanno ottenuto prestazioni all'avanguardia nei compiti di stima della posa e dello sguardo. Allo stesso modo, sono stati ottenuti buoni risultati per la stima e la previsione dello sguardo utilizzando un'architettura Gan spazio- temporale.

## 2) Sintesi delle immagini

Gran parte della recente ricerca su GAN si concentra sul miglioramento della qualità e dell'utilità delle funzioni di generazione delle immagini. Un esempio è il modello LAPGAN che introduce una cascata di reti convoluzionali all'interno di una struttura piramidale laplaciana per generare immagini da grossolane a fini. Questo modello estende la versione condizionale del modello GAN in modo che entrambe le reti G e D ricevono in ingresso informazioni aggiuntive sull'etichetta; questa tecnica si è dimostrata utile ed è ormai una pratica comune per migliorare la qualità dell'immagine. Questa idea del condizionamento GAN è stata successivamente estesa per incorporare il linguaggio naturale. Ad esempio, è stata utilizzata un'architettura GAN per sintetizzare immagini da descrizioni testuali, che potevano essere descritte come didascalie inverse. Data la didascalia testuale di un uccello come "bianco con un po' di nero sulla testa e sulle ali e un lungo becco arancione", il GAN addestrato poteva generare diverse immagini plausibili che corrispondevano alla descrizione. Così, oltre a condizionare le descrizioni del testo, la Generative Adversarial What-Where Network (GAWWN) condiziona la posizione delle immagini. Il sistema GAWWN supporta un'interfaccia interattiva in cui le immagini di grandi dimensioni possono essere costruite in modo incrementale con descrizioni testuali delle parti e riquadri di delimitazione forniti dall'utente. I GAN condizionali non solo ci consentono di sintetizzare nuovi campioni con attributi specifici, ma anche di sviluppare strumenti intuitivi per modificare le immagini, ad esempio per cambiare l'acconciatura di una persona in un'immagine, farle indossare gli occhiali o farla sembrare più giovane.

## 3) Traduzione da immagine a immagine

Le reti condizionali avversarie sono adatte a trasformare un'immagine di ingresso in un'immagine di uscita, un tema ricorrente nella grafica computerizzata, nell'elaborazione delle immagini e nella visione artificiale. Il modello pix2pix offre una soluzione generale a questa serie di problemi; non solo imparano una mappatura dall'immagine di ingresso a quella di uscita, ma costruiscono anche una funzione di perdita per imparare questa mappatura. Il modello ha mostrato risultati efficaci per diversi problemi di visione computerizzata che in precedenza richiedevano macchine separate, tra cui la segmentazione semantica, la generazione di mappe da fotografie aeree e la colorazione di immagini in bianco e nero.

## **Multi-Agent Reinforcement Learning**

Un ambiente multi-agente in un sistema di intelligenza artificiale è un ambiente operativo caratterizzato dalla presenza di due o più agenti razionali. In queste condizioni di ambiente ciascun agente razionale deve tenere in conto sia delle condizioni dell'ambiente e sia del comportamento degli altri agenti. I sistemi multi-agente possono essere utilizzati per affrontare problemi in una varietà di settori, tra cui la robotica, il controllo distribuito, le telecomunicazioni e l'economia. La complessità dei compiti che sorgono in questi settori li rende difficili da risolvere con comportamenti pre-programmati. Gli agenti devono invece scoprire una soluzione da soli, utilizzando l'apprendimento. Una parte significativa della ricerca sull'apprendimento multi-agente riguarda le tecniche di reinforcement learning.

Il reinforcement learning riveste un ruolo cruciale nei sistemi multi-agente, potenziando le capacità decisionali e di apprendimento degli individui coinvolti. Tramite iterazioni di tentativi ed errori, gli agenti assimilano feedback dall'ambiente o da altri agenti, sotto forma di ricompense o penalità, per perfezionare le proprie strategie.

Immaginate una flotta di veicoli autonomi che solcano le strade di una città affollata. Ogni veicolo si propone di raggiungere la destinazione il più rapidamente possibile evitando collisioni. Grazie al reinforcement learning, questi veicoli apprendono dai loro percorsi. Una valutazione positiva, in termini di ricompensa, è assegnata al raggiungimento della destinazione senza incidenti, mentre una valutazione negativa, tramite penalità, si verifica in caso di violazioni del codice stradale o di incidenti. Nel tempo, gli agenti si adattano, affinando le loro decisioni e migliorando il flusso del traffico.

Il reinforcement learning nei sistemi multi-agente diventa ancor più interessante quando gli agenti non imparano solo dall'ambiente, ma anche dalle azioni dei loro coetanei. Questo introduce il concetto di apprendimento in un contesto competitivo o cooperativo, in cui le strategie degli agenti si modulano in base all'osservazione reciproca. Attraverso questo scambio dinamico, gli agenti sviluppano tattiche e strategie sofisticate, generando comportamenti emergenti non precedentemente programmati.

Dato che, in un sistema multi-agente, gli agenti interagiscono con altri agenti, possiamo avere diversi tipi di ambienti:

- Ambienti cooperativi: dove gli agenti devono massimizzare i benefici comuni.

Ad esempio, in un magazzino, i robot devono collaborare per caricare e scaricare i pacchi in modo efficiente.

- Ambienti competitivi: in questo caso, l'agente desidera massimizzare i propri benefici minimizzando quelli dell'avversario.

Per esempio, in una partita di tennis, ogni agente vuole battere l'altro agente.

- Un mix di entrambi: due agenti fanno parte di una squadra (blu o viola): devono cooperare tra di loro e battere la squadra avversaria.

In modo simile al reinforcement learning per un singolo agente, il reinforcement learning per agenti multipli è modellato come una forma di processo decisionale di Markov (MDP). Ad esempio:

- Un insieme  $S$  di stati dell'ambiente.
- Un insieme  $A_i$  di azioni per ciascuno degli agenti  $i \in I = \{1, \dots, N\}$ .
- $P_{\vec{a}}(s, s') = \Pr(st+1 = s' | st = s, \vec{a}t = \vec{a})$  è la probabilità di transizione (al tempo  $t$ ) dallo stato  $s$  allo stato  $s'$  sotto l'azione congiunta  $\vec{a}$ .
- $R_{\vec{a}}(s, s')$  è la ricompensa congiunta immediata dopo la transizione da  $s$  a  $s'$  con l'azione congiunta  $\vec{a}$ .

Attraverso l'approccio del Multi-Agent Reinforcement Learning (MARL), gli agenti possono apprendere e comunicare tra di loro, portando a un miglioramento dell'efficienza nell'esecuzione dei compiti e nella presa decisionale. Sebbene l'utilizzo del reinforcement learning sia ancora in fase di valutazione per i sistemi di produzione, diverse applicazioni industriali sembrano promettenti per l'adozione di questa tecnologia:



**Controlli avanzati:** Affrontare il controllo dei sistemi non lineari è una sfida complessa, spesso risolta linearizzando il sistema in vari punti operativi. Il reinforcement learning può essere direttamente applicato ai sistemi non lineari, migliorando il controllo.

**Guida autonoma:** Nella decisione di guida basata sull'input visivo, il reinforcement learning si dimostra efficace, sfruttando il successo delle reti neurali profonde nell'analisi delle immagini.

**Robotica:** Il reinforcement learning può migliorare le prestazioni dei robot in applicazioni come il grasping, insegnando a manipolare diversi oggetti per operazioni di pick-and-place. Altre applicazioni includono la collaborazione tra uomini e robot, e tra robot stessi.

**Pianificazione:** I problemi di pianificazione compaiono in vari contesti, come il controllo dei semafori e la gestione delle risorse in fabbrica. Il reinforcement learning emerge come alternativa efficace ai metodi evolutivi per risolvere problemi di ottimizzazione combinatoria.

**Calibrazione:** Applicazioni che richiedono la calibrazione manuale dei parametri, come le unità di controllo elettronico (ECU), possono trarre vantaggio dall'implementazione del reinforcement learning.

Oltre ai vantaggi come la velocizzazione resa possibile dalla computazione parallela, diversi agenti di reinforcement learning possono trarre nuovi benefici dalla condivisione dell'esperienza, ad esempio attraverso la comunicazione, l'insegnamento o l'imitazione. Al contrario, sorgono diverse nuove sfide nel reinforcement learning per agenti multipli: la difficoltà di specificare un obiettivo di apprendimento, la non-stazionarietà del problema di apprendimento e la necessità di coordinazione.

La condivisione dell'esperienza può aiutare gli agenti di reinforcement learning con compiti simili ad apprendere più rapidamente e raggiungere prestazioni migliori. Ad esempio, gli agenti possono scambiarsi informazioni tramite la comunicazione, gli agenti esperti possono fungere da insegnanti per gli apprendisti, o gli apprendisti possono osservare e imitare gli agenti esperti. Quando uno o più agenti falliscono in un sistema multi-agente, gli agenti rimanenti possono assumere alcuni dei loro compiti. Ciò implica che il reinforcement learning per agenti multipli è intrinsecamente robusto. Inoltre, per design la maggior parte dei sistemi multi-agente permette anche l'inserimento facile di nuovi agenti nel sistema, portando a un alto grado di scalabilità.

Specificare un obiettivo MARL valido nel gioco stocastico generale è una sfida difficile, perché i rendimenti degli agenti sono correlati e non possono essere massimizzati indipendentemente. Diversi tipi di obiettivi MARL sono stati proposti nella letteratura, che tengono conto della stabilità della dinamica di apprendimento dell'agente, dell'adattamento al cambiamento del comportamento degli altri agenti, o sia della stabilità che dell'adattamento.

La non-stazionarietà si presenta nel reinforcement learning per agenti multipli perché tutti gli agenti nel sistema stanno imparando contemporaneamente. Ogni agente si trova quindi di fronte a un problema di apprendimento di un bersaglio mobile: la migliore politica cambia man mano che cambiano le politiche degli altri agenti.

La necessità di coordinazione deriva dal fatto che l'effetto dell'azione di qualsiasi agente sull'ambiente dipende anche dalle azioni intraprese dagli altri agenti. Di conseguenza, le azioni scelte dagli agenti devono essere reciprocamente consistenti al fine di raggiungere il loro effetto desiderato. Si consideri, ad esempio, che diversi paesi hanno reti elettriche interconnesse, e ciascuna rete di un paese è gestita da un agente. Anche se l'obiettivo principale di ciascun agente è ottimizzare gli interessi energetici del proprio paese, gli agenti devono comunque coordinarsi sui flussi di energia tra paesi vicini per ottenere una soluzione significativa.

Esistono diverse tecniche utilizzate nel MARL per affrontare le sfide sopra menzionate. Alcuni esempi includono:

1. Approcci basati su Q-learning distribuito: Estendono l'algoritmo di Q-learning per il contesto multi-agente, permettendo agli agenti di apprendere politiche di azione basate sull'esperienza passata.
2. Approcci basati su algoritmi genetici: Utilizzano l'evoluzione darwiniana per addestrare politiche di comportamento per gli agenti, sfruttando il concetto di selezione naturale e mutazione.
3. Approcci basati su apprendimento federato: Gli agenti condividono conoscenze e politiche di apprendimento tramite una rete federata, consentendo un apprendimento collaborativo e distribuito.
4. Approcci basati su sistemi multi-agente emergenti: Modellano gli agenti come parte di un sistema più ampio, studiando i fenomeni emergenti che si verificano quando gli agenti interagiscono tra loro.

Sebbene le sfide del reinforcement learning per agenti multipli siano state ampiamente studiate e valutate, le metodologie esistenti per acquisire comportamenti multi-agente non riescono a sfruttare appieno le molteplici opportunità all'interno di un sistema multi-agente. Nonostante progressi sostanziali, particolarmente in aree uniche di apprendimento in un sistema multi-agente, come la modellazione degli agenti, il trasferimento di conoscenza, il gioco di squadra ad hoc, l'assegnazione di crediti e le sfide fondamentali del reinforcement learning per agenti multipli, esiste un potenziale inutilizzato che richiede ulteriori esplorazioni e perfezionamenti.

## Support Vector Machine

Le macchine a vettori di supporto, o SVM (Support Vector Machine), sono un tipo di algoritmo di apprendimento supervisionato utilizzato per la classificazione e la regressione. Si tratta di un classificatore binario e non probabilistico.

L'aspetto non probabilistico è in contrasto con i classificatori probabilistici che invece calcolano la probabilità di appartenenza alla classe sulla base degli esempi di addestramento.

Il support vector machine presenta l'obiettivo di trovare un iperpiano che separi in modo ottimale i dati di input in diverse classi. In particolare, i dati possono essere divisi in un massimo di due classi, in quanto la divisione di dati multi-classe con SVM è ancora un'area di ricerca attiva. I support vector, ovvero i vettori di supporto, sono i punti più vicini all'iperpiano e tali punti dipendono dal set di dati che si sta analizzando e, se, vengono rimossi o modificati, alterano la posizione dell'iperpiano divisorio.

Visto che si tratta di un classificatore binario un SVM rappresenta ogni oggetto dati come un punto nello spazio dimensionale delle caratteristiche, che appartiene a una sola delle due classi.

Ad ogni classe viene assegnata un'etichetta che può essere  $y_i=1$  o  $y_i=-1$ . Quindi la rappresentazione matematica del set di dati è:

$$\text{Data} = \{(x_i, y_i) / x_i \in \mathbb{R}^p, y_i \in (-1, +1)\}_{i=1}^n$$

P è la dimensione del vettore della caratteristica;

$n$  è il numero di vettori;

$\mathcal{R}$  è lo spazio dimensionale;

L'etichetta risulta fondamentale in fase di addestramento in quanto il SVM costruisce un modello che assegna i nuovi esempi a una delle due classi. Gli esempi vengono rappresentati come punti nello spazio e la classe di appartenenza viene individuata tenendo in considerazione il lato su cui ricadono i punti stessi.

Le due classi risultano divise da un iperpiano lineare che meglio separa le diverse classi nello spazio delle funzionalità. Questo iperpiano viene scelto in modo che il margine tra l'iperpiano e i punti più vicino ad esso, noti come vettori di supporto, sia massimizzato. Il margine non è altro che la distanza minima dalla retta ai punti delle due classi e l'aspetto interessante è che maggiore è tale margine, migliore sarà la generalizzazione. Il motivo è molto semplice, infatti tanto più grande sarà il margine tra le due classi tanto minore sarà la possibilità di fare confusione.

Non sarebbe possibile, infatti, una rappresentazione non lineare in quanto quest'ultima si adatterebbe troppo strettamente a un confine che separerebbe perfettamente i dati di addestramento; questo fenomeno si chiama *overfitting*. La conseguenza sarebbe un modello che ha scarsa generalizzazione prevedendo in modo impreciso la classe dei nuovi dati.

Per evitare l'*overfitting* si tiene in considerazione un iperpiano lineare che è rappresentato dalla seguente equazione:

$$w \bullet x = 0$$

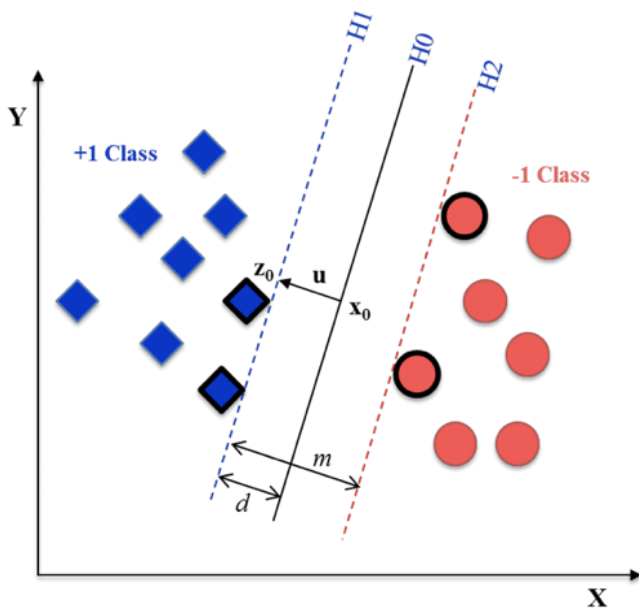
Essa rappresentata nello spazio dimensionale non sarà altro che una linea  $y=ax+b$ .

Il vettore  $w$  definisce una linea specifica che graficamente è l'iperpiano lineare; pertanto, qualsiasi punto definito dal vettore della caratteristica  $(x, y)$ , deve trovarsi da qualche parte su quella linea.

Dato che i vettori sono vettori colonna, il loro prodotto scalare equivale all'operazione di matrice  $w^T x = 0$ .

Le forme vettoriali e matriciali sono più facili da rappresentare nella memoria del computer per calcoli più rapidi.

Ritornando al nostro modello di riferimento, ovvero SVM, ricordiamo che il suo obiettivo è quello di trovare una retta che separa le due classi di dati e soprattutto che massimizza il margine tra le classi stesse. La massimizzazione del margine avviene utilizzando una parte minima del dataset di addestramento, i cosiddetti vettori di supporto, che risultano evidenziati nel grafico riportato in seguito.



Come si può osservare nel grafico, la regione delimitata dai due iperpiani H1 e H2 non ha punti dati, è una regione vuota; pertanto, H0 rappresenta il confine decisionale in quanto è l'iperpiano lineare che massimizza la distanza tra i punti delle due classi.

Il margine dell'iperpiano è invece la distanza di separazione tra i due iperpiani H1 e H2.

I cerchi e rombi evidenziati in nero (vettori di supporto), sono gli unici esempi del dataset che risiedono sul margine dell'iperpiano.

Una volta individuati questi, tutti gli altri vettori sono insignificanti ai fini della classificazione, perché sono loro a definire la retta di separazione e il margine, visto che sono quelli più vicini all'altra classe e dunque sono anche i valori più difficili da classificare.

Proprio questo è ciò che differenzia l'algoritmo SVM dagli altri algoritmi di classificazione perché, mentre una regressione logistica impara a classificare prendendo come riferimento gli esempi più rappresentativi della classe, la SVM cerca gli esempi più difficili, quelli più vicini all'altra classe (appunto, i vettori di supporto), e considera solo quelli per la classificazione.

L'algoritmo SVM può anche essere spiegato come se fosse un gioco dove abbiamo due attori in competizione tra di loro. In particolare, un giocatore sfida l'avversario a trovare il miglior iperpiano che gli dia i punti più difficili da classificare.

La soluzione del gioco sarà rappresentata da un compromesso tra le loro abilità strategiche: infatti, il primo giocatore dovrà essere bravo a sfidare il secondo nel trovare i punti più difficili, mentre il secondo dovrà essere abile a identificare i limiti della decisione migliore.

Considerato che le SVM sono molto apprezzate per la loro capacità di gestire una grande quantità di dati, questo le rende utilizzabili in diversi campi come:

- Il riconoscimento di immagini;
- La classificazione del testo;
- L'analisi del sentiment;
- La bioinformatica.

### **L'importanza della Connessione tra Intelligenza Artificiale e Teoria dei Giochi**

L'interconnessione tra intelligenza artificiale (IA) e teoria dei giochi rappresenta un campo di studio cruciale e altamente promettente, che offre vantaggi significativi sia teorici che pratici. La teoria dei giochi fornisce un quadro matematico per analizzare le interazioni strategiche tra agenti razionali, mentre l'IA utilizza algoritmi avanzati per simulare e ottimizzare tali interazioni in ambienti complessi. In sintesi, l'integrazione tra teoria dei giochi e intelligenza artificiale non solo arricchisce entrambi i campi, ma crea anche nuove opportunità per affrontare problemi complessi in modo più efficiente e innovativo. Questa sinergia promette di trasformare settori critici della nostra società, migliorando la capacità dei sistemi di IA di prendere decisioni informate, adattarsi a nuovi contesti e collaborare in maniera efficace.

Redatta da:

Mariateresa Giordano

Cristina Ligato

Maria Lo Giudice

Alfonso Longo

Jessica Manglaviti

Marika Manglaviti

Eliana Moraca

Irene Nevolo

Anna Ripepi

Alessia Sinicropi

Rocco Christian Sorrenti

Valeria Vadalà

William Valenti