Algorithmic Game Theory

Daniele Avolio

A.A. 2023/2024

Contents

1 Introduzione

Definizione esame: Solitamente lo schema delle lezioni sarà

$$LezioneTeoria \implies LezioneLaboratorio \tag{1}$$

La lezione di Lab sarà fatta praticamente spesso in *Python*.

2 Cos'è la Game Theory — Teoria dei giochi

La **teoria dei giochi** è una disciplina che studia il comportamento decisionale multi-persona, usato per fare predizioni su come **agenti razionali multipli** interagiscono o si comportano in situazioni di *cooperazione* o in situazione di *conflitto*.

Alcune definizioni di termini:

- **Conflitto**: le azioni dei giocatori hanno effetto sugli Algorithmic
- **Cooperazione**: I giocatori possono collaborare per raggiungere un obiettivo
- **Comportamento razionale**: I giocatori vogliono massimizzare la loro *utilità attesa — expected utility*
- **Predizione**: Il nostro obiettivo è sapere cosa faranno i giocatori, utilizzando *solution concepts - concetti di soluzione*

2.1 E cosa significa Algorithmic Game Theory?

Possiamo dire che algorithmic game theory è un punto d'incontro tra **game theory** e **algorithm design** che punta a ***progettare algoritmi** che permettono deglle strategie in specifici ambienti.*

2.2 Coalition Games

La **coalition game theory** è una branca della game theory che studia le interazioni tra gruppi di giocatori, che **collaborano** per *raggiungere un obiettivo comune*.

Nota - Shapley Values: Il concetto di **Shapley Values** è un concetto che permette di *spiegare*, circa, come un algoritmo di **machine learning** ha preso una decisione. Ad esempio, mostra le *feature* che hanno avuto un impatto maggiore nella decisione finale della predizione. In pratica mostra i vari *Join — Coalizioni* di features.

Quali sono le domande più importanti in questa sezione?

- Quale coalizione è più probabile che venga formata?
- In che modo i giocatori devono dividere il premio? *(Payoff)*

2.3 Non-Cooperative Games

In questo tipo di giochi, i giocatori **non hanno coalizioni** o comunque non ne hanno bisogno.

Alcuni giochi che fanno parte di questa categoria:

- Scacchi
- Sasso-Carta-Forbice
- *Il dilemma del prigioniero*

	Giocatore 1	
Giocatore 2	Collabora	Tradisci
Collabora	(-1,-1)	(-5,0)
Tradisci	(0,-5)	(-3,-3)

Figure 1: Esempio di dilemma del prigioniero

In questo gioco, la **strategia** migliore per il singolo è quella di **tradire** l'altro giocatore, in quanto è quella che massimizza la sua utilità, precisamente andrebbe a **perdere 0 punti**, mentre l'altro giocatore ne perderebbe 5.

2.4 Tree Decomposition

Alcuni problemi sui grafi hanno una complessità di **NP-HARD** su dei grafi arbitrari, e hanno bisogno di alcune soluzioni che avranno implementazioni complesse e **programmazione dinamica.**

2.5 Computational Social Choice

Questa sezione parla di computazione di risultati risultanti da **regole di voto** — **voting rules** e quali problemi ci possono essere nel rappresentare le preferenze dei giocatori.

	Verdetto		
	Evidenza1	Evidenza2	Colpevole
Giudice1	1	0	Innocente
Giudice2	0	1	Innocente
Giudice3	1	1	Colpevole

Figure 2: Esempio di votazione

Maggiore è il numero di persone che votano, maggiore è la probabilità che il risultato sia corretto.

2.6 Mechanism Design

E' un tipo di **reverse game theory**. Invece di analizzare come i giocatori si comportano in un gioco, lo scopo del *mechanism design* è quello di **creare un gioco** per portare i giocatori a ***comportarsi in un modo specifico*** che *vogliamo noi.* Un esempio molto semplice è il *maccanismo di asta di Ebay.* Altro esempio è quello dei *carrelli dei supermecati.* Il fatto di dover utilizzare una moneta per utilizzare il carrello **porta la persona** a dover riportare il carrello nello stesso posto, invece di lasciarlo in un luogo qualsiasi del supermercato.

2.7 Fair Division of Indivisible Goods

In questa sezione si parla di come dividere delle risorse in modo **fair** tra i giocatori. Ok?

2.8 Cake Cutting

In questa sezione si parla di come dividere dei **beni continui** in bsae alle *preferenze dei giocatori*.

1. Fairness
2. Proportionality
3. Envy-freeness

3 Giochi di Coalizione e Concetti di Soluzione

Spesso la **teoria dei giochi** fa riferimento a tipologie di giochi in cui gli agenti **non collaborano**, ma **competono** tra loro. In questi casi si parla di **gioco non cooperativo**.

Parliamo di un ambiente in cui degli agenti interagiscono tra loro, e ogni agente ha un suo **obiettivo** da raggiungere.

Definizione 3.1 (Gioco non cooperativo) *Un gioco non cooperativo è un gioco in cui gli agenti non collaborano tra loro.*

- *Un set di agenti* $N = \{1, \ldots, n\}$.
- *Ogni agente* $i \in N$ *ha un set di azioni* S
- *Ogni agnete* $i \in N$ *ha una funzione di utilità* $u_i : S_1 \times S_2 \cdots S_n \to \mathcal{R}$

Definizione 3.2 (Gioco cooperativo) *Il contrario di giochi non cooperativi sono, banalmente,* ***i giochi cooperatvi****, in cui gli agenti* ***collaborano*** *tra loro.*

Domanda: In quale caso le coalizioni appaiono nella teoria dei giochi cooperativi?

- Allocazioni di task
- Allocazione di risorse
- Esperienza degli agenti complementare tra loro

Esempio di gioco cooperativo: Immaginiamo di avere 9 agenti. Ora, gli agenti devono sceliere:

- Con chi allearsi
- Come agire
- Come dividere il premio

Immaginiamo di avere $p_1, p_2, p_3, c_1, c_2, c_3, e_1, e_2, e_3$. Immaginiamo queste 3 coalizioni:

- $C_1\{c_1, e_3, p_3\}$
- $C_2\{c_3, e_2, p_2\}$
- $C_3\{c_2, e_1, p_1\}$

Una **struttura di coalizione** è del tipo: $CS =< C_1, C_2, C_3 >$.

Definiamo anche **il vettore azioni** $a =< a_{c_1}, a_{c_2}, a_{c_3} >$.

Allora possiamo avere un'**allocazione di risorse**

$$u(C_3|a_{c_3}) = 30 \implies \textit{Allocazione} :< p_1 = 12, c_2 = 3, e_1 = 15 > \tag{2}$$

Quindi, diciamo che nei giochi collaborativo:

- I giocatori **formano coalizioni**
- Ogni coalizione ha associato un **worth**
- Alla fine c'è un **total worth** da distribuire

3.0.1 Tassonomia dei giochi cooperativi

Quando parliamo di giochi cooperativi parliamo di giochi in cui i giocatori tra loro collaborano, fanno azioni insiee, e si formano dei vincoli tra loro. Ma dobbiamo differenziare due tipi di **utility games**

Definizione 3.3 (Transferable Utility Games) *La paga viene data al gruppo e si divide tra loro.*

Definizione 3.4 (Non Transferable Utility Games) *L'azione del gruppo fornisce la paga ai singoli giocatori in modo individuale.*

Esempio di Transferable Utility Games:

Hai N bambini, ognuno dei quali ha una certa quantità di denaro: il bambino i-esimo ha b_i dollari.

Sono in vendita tre tipi di vaschette di gelato:

- Tipo 1 costa $7 e contiene 500g.
- Tipo 2 costa $9 e contiene 750g.
- Tipo 3 costa $11 e contiene 1kg.

I bambini hanno una preferenza per il gelato e non si preoccupano del denaro.

Il risultato ottenuto da ciascun gruppo è la quantità massima di gelato che i membri del gruppo possono acquistare unendo il loro denaro. Il gelato può essere condiviso liberamente all'interno del gruppo.

Formalizzazione dei giochi cooperativi:

Un gioco di utilità trasferibile è una coppia (N, v), dove:

- $N = \{1, \dots, n\}$ è l'insieme dei giocatori (anche chiamato coalizione grandiosa).
- $v : 2^N \to \mathbb{R}$ è la funzione caratteristica.
- Per ogni sottoinsieme di giocatori C, $v(C)$ è l'importo che i membri di C possono guadagnare lavorando insieme.

Facciamo delle assunzioni. Solitamente diciamo che v è **normalizzato**, cioé $v(\emptyset) = 0$. Ci sono altri due casoi però:

- **Non-negativo**: $v(C) \geq 0$ per ogni $C \subseteq N$.
- **Monotono**: $v(C) \leq v(D)$ per ogni C, D t.c $C \subseteq D$.

Tutto questo non è sempre uguale e dipende sempre dallo scenario.
Esempio del gioco dl gelato: Abbiamo tre giocatori:

1. C con 6
2. M con 4
3. P con 4

Ora, abbiamo 3 tipi di gelato con :

- Gelato 1: w = 500 e p = 7
- Gelato 2: w = 750 e p = 9
- Gelato 3: w = 1000 e p = 11

Cosa possiamo dire? Innanzitutto, **nessuno può comprare niente da solo.** Quindi, se vogliamo che qualcuno compri qualcosa, dobbiamo formare una coalizione. Le domande da fare sono: *Quali azioni dobbiamo compiere? In quale modo ci dividiamo il premio?.*

$$\begin{aligned} v(\emptyset) = v(\{C\}) = v(\{M\}) = v(\{P\}) = 0 \\ v(\{C, M\}) = 750 \\ v(\{C, P\}) = 750 \\ v(\{M, P\}) = 500 \\ v(\{C, M, P\}) = 1000 \end{aligned}$$

Definizione 3.5 (Outcome) *Un outcome (o risultato) di un gioco di utilità trasferibile $G = (N, v)$ è una coppia (CS, x), in cui:*

- $CS = (C_1, \dots, C_k)$ *è una struttura di coalizione, cioè una partizione di N.*
- $\bigcup_i C_i = N$, $C_i \cap C_j = \emptyset$ *per* $i \neq j$.
- $x = (x_1, \dots, x_n)$ *è un vettore di pagamento che distribuisce il valore di ciascuna coalizione in CS.*
- $\sum_{i \in C} x_i = v(C)$ *per ogni C in CS (Efficienza).*

Supponiamo che $v(\{1,2,3\}) = 9$ *e* $v(\{4,5\}) = 4$.
Quindi, $((1,2,3,4,5),(3,3,3,3,1))$ *è un* ***risultato***.
Invece, $((1,2,3,4,5),(2,3,2,3,3))$ *non è un risultato.*
I ***trasferimenti tra coalizioni*** *non sono consentiti. Un risultato* $(CS, \underline{x})$ *è chiamato* ***imputazione*** *se soddisfa la* ***razionalità individuale****:* $x_i \geq v(\{i\})$ *per tutti* $i \in N$.

Definizione 3.6 (Giochi superadditivi) *Un gioco di utilità trasferibile* $G = (N, v)$ *è chiamato superadditivo se* $v(C \cup D) \geq v(C) + v(D)$ *per qualsiasi due coalizioni disgiunte* C *e* D.
Esempio: $v(C) = |C|^2$*;* $v(C \cup D) = (|C|+|D|)^2 \geq |C|^2 + |D|^2 = v(C) + v(D)$.
Nei ***giochi superadditivi****, due coalizioni possono sempre fondersi senza perdere denaro; quindi, possiamo assumere che i giocatori formino la coalizione grandiosa.*
Praticamente, quando due coalizioni collaborando ottengono un risultato che è almeno pari alla somma dei risultati che avrebbero ottenuto se avessero agito da sole.
Un esempio è il seguente:

$$
\begin{aligned}
v(\emptyset) = v(\{C\}) &= v(\{M\}) = v(\{P\}) = 0 \\
v(\{C, M\}) &= 750 \\
v(\{C, P\}) &= 750 \\
v(\{M, P\}) &= 500 \\
v(\{C, M, P\}) &= 1000
\end{aligned}
$$

In questo caso si vede che è un gioco superadditivo perché la collaborazione porta ad un risultato migliore.

3.1 Concetti di soluzione

Assumiamo che la grande coalizione **N** sia formata. Quindi:

$$x = (x_1, x_2, \dots, c_n)$$

$$x_i \geq 0 \forall i \in N$$

$$x_1 + x_2 + \dots x_n = v(N)$$

Considera il gioco del gelato con la seguente funzione caratteristica. Si tratta di un gioco superadditivo in cui gli esiti sono vettori di pagamento (modi per dividere 1000).

Come dovrebbero i giocatori condividere il gelato?

Se lo dividono come (200, 200, 600), Charlie e Marcie possono ottenere più gelato acquistando una vaschetta da 750g da soli e dividendo equamente. L'esito (200, 200, 600) **non è stabile**!

Definizione 3.7 (Core O Nucleo) *Il nucleo o core di un gioco è l'insieme di tutti gli* ***esiti che sono stabili****, cioé quegli esiti che no vengono scartati da nessuna coalizione.*

$$core(G) = \{(CS, \underline{x})| \sum_{i \in C} x_i \geq v(C) \ for \ any \ C \subseteq N\}$$

Nota: **x(c)** si identifica come la somma dei valori $\sum_{i \in C} x_i$.
Possiamo accorciare in:

- $x \in R^n$ è un nucleo se $x(c) \geq v(c) \forall C \subseteq N$
- x(N) = v(N)

Torniamo al nostro esempio. Ora vediamo il concetto di **nucleo** applicato

- (200, 200, 600) **non è** nel nucleo:
- $v(\{C, M\}) > x_C + x_M$
- (500, 250, 250) **è nel** nucleo:

nessun sottogruppo di giocatori può deviare in modo che ciascun membro del sottogruppo ottenga di più. Un vettore (x_C, x_M, x_P) è nel nucleo se e solo se soddisfa le seguenti condizioni:

- $x_C + x_M \geq v(\{C, M\})$ (stabilità)
- $x_C + x_P \geq v(\{C, P\})$ (stabilità)
- $x_P + x_M \geq v(\{P, M\})$ (stabilità)
- $x_C \geq v(\{C\})$ (razionalità individuale)
- $x_P \geq v(\{P\})$ (razionalità individuale)
- $x_M \geq v(\{M\})$ (razionalità individuale)
- $x_C + x_P + x_M = v(\{C, M, P\})$ (efficienza)

Queste **3 proprietà** sono importanti.

Definizione 3.8 (Giochi con nucleo vuoto) *Il concetto di nucleo è molto attraente come concetto di soluzione. Purtroppo, ci sono alcuni giochi che hanno* ***un nucleo vuoto****.*

Consideriamo il gioco $G = (\{1, 2, 3\}, v)$ con la seguente funzione caratteristica v:

$$v(C) = \begin{cases} 1 & se \ |C| > 1 \\ 0 & altrimenti \end{cases}$$

Considera un risultato $(CS, \underline{x})$:
Supponi che $CS = (\{1\}, \{2\}, \{3\})$.

- In questo caso, la coalizione grandiosa può deviare.
- $x_1 + x_2 + x_3 = v(\{1\}) + v(\{2\}) + v(\{3\}) < v(\{1, 2, 3\})$.

Cioè, **non è stabile**. Guarda la disequazione.
Supponi che $CS = (\{1, 2\}, \{3\})$.

- In questo caso, o 1 o 2 ottengono meno di 1, quindi possono deviare con 3.
- $x_1 + x_3 = x_1 + 0 < 1 < v(\{1, 3\})$.

Collaborando devono **dividere** questo **1** che ottengono. Quindi, 1 o 2 prenderò meno dell'altro e l'altro potrebbe deviare con il 3.
Supponi che $CS = (\{1, 2, 3\})$.

- In questo caso, $x_i > 0$ vale per alcuni i, diciamo $i = 3$.
- Quindi, $x(\{1, 2\}) < 1$, ma $v(\{1, 2\}) = 1$.

Quindi in ogni caso *qualcuno potrebbe cercare una coalizione migliore di quella in cui si trova attualmente*. Questo porta ad avere un **nucleo vuoto**.

3.1.1 Cosa fare quando il nucleo è vuoto?

Questa situazione prende il nome di $\epsilon - Core$. In questo casi si vuole approssimare un esito stabile.

Bisogna *rilassare* il concetto di nucleo:

- Nucleo: $(CS, x) : x(C) \geq v(C)$ per ogni $C \subseteq N$
- ϵ-nucleo: $(CS, x) : x(C) \geq v(C) - \epsilon$ per ogni $C \subseteq N$

Solitamente questa nozione è definita solo per giochi superadditivi.

Per esempio, consideriamo il gioco $G = (\{1, 2, 3\}, v)$, con $v(C) = 1$ se $|C| > 1$, $v(C) = 0$ altrimenti:

- Il $\frac{1}{3}$-nucleo non è vuoto: $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) \in \frac{1}{3}$-nucleo
- Il ϵ-nucleo è vuoto per qualsiasi $\epsilon < \frac{1}{3}$:
- – $x_i \geq \frac{1}{3}$ per qualche $i = 1, 2, 3$; quindi $x(N\{i\}) \leq \frac{2}{3}$, ma $v(N\{i\}) = 1$.

Definizione 3.9 (Nucleo minimo) *Definiamo $\epsilon^*(G)$ come* $\inf\{\epsilon | \epsilon\text{-}core\ di\ G\ non\ è\ vuoto\}$.

- *Si può dimostrare che $\epsilon^*(G)$-core non è vuoto.*

La definizione di $\epsilon^*(G)$-core è il nucleo minimo di G.

- $\epsilon^*(G)$ *è chiamato il valore del nucleo minimo.*

Nel contesto del gioco $G = (\{1, 2, 3\}, v)$ con la funzione caratteristica $v(C)$ definita come segue:

- $v(C) = 1$ se $|C| > 1$
- $v(C) = 0$ altrimenti
- Il 1/3-core non è vuoto: $(1/3, 1/3, 1/3) \in 1/3$-core
- Il ϵ-core è vuoto per qualsiasi $\epsilon < 1/3$:
  - $x_i \geq 1/3$ per qualche $i = 1, 2, 3$, quindi $x(N\{i\}) \leq 2/3$, ma $v(N\{i\}) = 1$.

3.2 Concetti di soluzione avanzati

Ci sono in particolare due che sono molto importanti: **shapy value** e il **Nucleolus**

Più sofisticate considerazioni sulla stabilità

- **Nucleolus**: Il nucleolus è un concetto utilizzato nella teoria dei giochi cooperativi per valutare la giustizia nella distribuzione dei guadagni tra i giocatori.
- **Bargaining set**: L'insieme di contrattazione è un concetto che si riferisce agli insiemi di risultati in cui i giocatori trovano equo e ragionevole partecipare, dati i poteri di contrattazione.
- **Kernel**: Il nucleo è un sottoinsieme del nucleo in cui i giocatori non possono migliorare il proprio risultato cooperando in modo diverso.

Concetto di fairness

- **Shapley value**: Il valore di Shapley è una soluzione per assegnare un valore a ciascun giocatore in modo equo, tenendo conto del loro contributo marginale a ogni possibile coalizione.
- **Banzhaf index**: L'indice di Banzhaf è una misura del potere di voto di ciascun giocatore in un gioco di voto ponderato.

3.2.1 Nucleolus

Definiamo ora il concetto di *eccesso*

Definizione 3.10 (Eccesso) *E' una misura che indica quanto la coalizione* ***è insoddisfatta.***

$$e(S, x) = v(S) - x(S) \tag{3}$$

Facciamo un esempio numerico:

- v({1}) = v({2}) = v({3}) = 0
- v({1,2}) = v({1,3}) = v({2,3}) = 1
- v({1,2,3}) = 3

Minimizzare la insoddisfazione di ogni possibile coalizione. Applichiamo

- x = (0,0,3) $\implies e(\{1,2\},x) = v(\{1,2\}) - (x_1 + x_2) = 1 - 0 = 1$
- x = (1,2,0) $\implies e(\{1,2\},x) = v(\{1,2\}) - (x_1 + x_2) = 1 - 3 = -2$

RIVEDERE PARTE SUL Nucleolus pagina 68 circa
Ritorniamo alla definizione di Nucleolus.
Definizione da Schmeidler: Il nucleolus $\mathcal{N}(\mathcal{G})$ di un gioco $\mathcal{G}$ è l'insieme:

$$\mathcal{N}(\mathcal{G}) = \{x \in \mathcal{X}(\mathcal{G}) | \nexists y \in \mathcal{X}(\mathcal{G})\ t.c\ \theta(y) \lessapprox DACMABIARE\theta(X)\}$$

3.3 Shapley Value

Stability vs Fairness

Consideriamo il gioco $G = (\{1,2\}, v)$ con le seguenti caratteristiche:

- $v(\emptyset) = 0$
- $v(\{1\}) = v(\{2\}) = 5$
- $v(\{1,2\}) = 20$

Nel nucleo del gioco, abbiamo l'allocazione $(15,5)$. In altre parole, il giocatore 1 riceve 15 e il giocatore 2 riceve 5. È importante notare che il nucleo rappresenta una situazione in cui nessun giocatore può ottenere un risultato migliore deviando unilateralmente. In questo caso, il giocatore 2 non può ottenere un risultato migliore deviando.

La domanda principale è: l'allocazione (15, 5) è equa? La giustizia in un contesto di gioco può essere soggettiva e dipendere dalle aspettative e dagli accordi tra i giocatori. Quindi, se l'allocazione è considerata equa o meno potrebbe variare in base al contesto e alle aspettative dei giocatori.

No! Since 1 and 2 are symmetric Outcomes in the core may be unfair! How do we divide payoffs in a fair way?

Pensiamo a questo. Un risultato equo dovrebbe premiare ciascun agente in base al loro contributo. Nel primo tentativo, dato un gioco $G = (N, v)$, impostiamo $x_i = v(\{1,\ldots,i-1,i\}) - v(\{1,\ldots,i-1\})$. In altre parole, il pagamento per ciascun giocatore è il loro contributo marginale alla coalizione dei loro predecessori. Otteniamo $x_1 + \ldots + x_n = v(N)$; x è un vettore di pagamento.

Tuttavia, questo metodo non funziona poiché il pagamento di ciascun giocatore dipende dall'ordine. Ad esempio, consideriamo il gioco $G = (\{1,2\}, v)$, con $v(\emptyset) = 0$, $v(\{1\}) = v(\{2\}) = 5$, e $v(\{1,2\}) = 20$. In questo caso, $x_1 = v(1) - v(\emptyset) = 5$ e $x_2 = v(\{1,2\}) - v(\{1\}) = 15$.

Notiamo che i risultati non sono gli stessi, indipendentemente dall'ordine. Pertanto, questa formulazione non produce risultati equi.

Un'idea per eliminare la dipendenza dall'ordine è quella di calcolare una media su tutte le possibili permutazioni degli ordini di arrivo.

Ad esempio, consideriamo il gioco $G = (\{1,2\}, v)$, con $v(\emptyset) = 0$, $v(\{1\}) = v(\{2\}) = 5$, e $v(\{1,2\}) = 20$. Iniziamo calcolando i pagamenti per due ordini diversi:

- Per l'ordine (1, 2): $x_1 = v(1) - v(\emptyset) = 5$ e $x_2 = v(\{1,2\}) - v(\{1\}) = 15$.
- Per l'ordine (2, 1): $y_2 = v(2) - v(\emptyset) = 5$ e $y_1 = v(\{1,2\}) - v(\{2\}) = 15$.

Ora, calcoliamo i pagamenti mediando tra i due ordini:

- $z_1 = (x_1 + y_1)/2 = (5 + 15)/2 = 10$
- $z_2 = (x_2 + y_2)/2 = (15 + 5)/2 = 10$

Il risultato ottenuto è equo, poiché ciascun giocatore riceve 10, indipendentemente dall'ordine in cui sono considerati.

Una permutazione di $\{1, \ldots, n\}$ è una corrispondenza uno a uno da $\{1, \ldots, n\}$ a se stessa.

Denotiamo con $P(N)$ l'insieme di tutte le permutazioni di N.

Denotiamo con $S_\pi(i)$ l'insieme dei predecessori di i in una permutazione $\pi \in P(N)$.

Per $C \subseteq N$, definiamo $\delta_i(C) = v(C \cup \{i\}) - v(C)$ come il contributo marginale del giocatore i a C.

Il valore di Shapley del giocatore i in un gioco $G = (N, v)$ con $|N| = n$ è dato da:

$$\varphi_i(G) = \frac{1}{n!} \sum_{\pi : \pi \in P(N)} \delta_i(S_\pi(i))$$

Supponiamo di scegliere una permutazione dei giocatori in modo uniforme e casuale tra tutte le possibili permutazioni di N. In questo contesto, il valore di Shapley φ_i rappresenta il contributo marginale atteso del giocatore i alla coalizione dei suoi predecessori.

♦ 3.3.1 Proprietà del valore di Shapley

Definizione 3.11 (Proprietà 1) *In qualsiasi gioco G, la somma dei valori di Shapley di tutti i giocatori, ossia $\varphi_1 + \ldots + \varphi_n$, è uguale al valore totale del gioco $v(N)$.*

Definizione 3.12 (Proprietà 2) *Un giocatore i è definito un giocatore fittizio (dummy) se $v(C) = v(C \cup \{i\})$ per qualsiasi insieme $C \subseteq N$.*

Proposizione: *Se un giocatore i è un giocatore fittizio, allora il suo valore di Shapley φ_i è uguale a 0.*

Definizione 3.13 (Proprietà 3) *Due giocatori i e j sono definiti simmetrici se $v(C \cup \{i\}) = v(C \cup \{j\})$ per qualsiasi insieme $C \subseteq N \setminus \{i, j\}$.*

Proposizione: *Se i e j sono giocatori simmetrici, allora i loro valori di Shapley φ_i e φ_j sono uguali.*

Definizione 3.14 (Proprietà 4) *Siano $G1 = (N, u)$ e $G2 = (N, v)$ due giochi con lo stesso insieme di giocatori. Allora $G = G1 + G2$ è il gioco con l'insieme di giocatori N e la funzione caratteristica w definita come $w(C) = u(C) + v(C)$ per tutti gli insiemi $C \subseteq N$.*

Proposizione: *Il valore di Shapley di un giocatore i nel gioco $G1 + G2$ è uguale alla somma dei valori di Shapley di i nei giochi $G1$ e $G2$, ossia $\varphi_i(G1 + G2) = \varphi_i(G1) + \varphi_i(G2)$.*

Proprietà riassunte:

1. **Efficienza:** $\varphi_1 + \ldots + \varphi_n = v(N)$
2. **Dummy:** Se i è un giocatore fittizio, allora $\varphi_i = 0$
3. **Simmetria:** Se i e j sono giocatori simmetrici, allora $\varphi_i = \varphi_j$
4. **Additività:** $\varphi_i(G1 + G2) = \varphi_i(G1) + \varphi_i(G2)$

Teorema: Il valore di Shapley è l'unico schema di distribuzione dei pagamenti che soddisfa le proprietà 1-4.

E' possibile scrivere la formula anche in questo modo: *da copiare dalle slide*

Appunti di Laboratorio

4 Introduzione alla teoria dell'utilità e decision making

4.1 Concetti di base

4.1.1 ALTERNATIVE

Parliamo di *agenti* che devono scegliere un'*alternativa* da un'insieme $\mathcal{X}$ di alternative. Questo insieme di alternative ha degli elementi che possono essere **esaustivi** o **mutualmente esclusivi**.

Esepmio: § {

- DL = Deep Learning
- AGT = Algorithmic Game Theory
- DLAGT = Deep Learning Algorithmic Game Theory
- N = None

}

4.1.2 PREFERENZE

Con il termine **preferenze** identifichiamo una relazione $\succcurlyeq$ su $\mathcal{X}$, che è un sottoinsieme di $\mathcal{X} \times \mathcal{X}$. Le preferenze possono essere:

- **complete** se $\forall x, y \in \mathcal{X}$ vale $x \succcurlyeq y$ oppure $y \succcurlyeq x$
- **transitive** se $\forall x, y, z \in \mathcal{X}$ vale $x \succcurlyeq y$ e $y \succcurlyeq z$ allora $x \succcurlyeq z$

4.1.3 Relazione di Preferenza

Una preferenze è una **relazione di preferenza** se è sia **completa** che **transitiva**.

Si chiama preferenza **stretta** se $x \succ y \iff x \succcurlyeq y$ e $x \not\preccurlyeq y$.

Si chiama **indifferenza** se $x \sim y \iff x \succcurlyeq y$ e $x \preccurlyeq y$.

4.1.4 Rappresentare le preferenze come utilità

Una relazione di preferenza puà essere tradotta in una funzione di utilità del tipo $u : \mathcal{X} \to \mathbb{R}$. Si può fare in questo modo:

$$x \succcurlyeq y \iff u(x) \geq u(y) \quad \forall x, y \in \mathcal{X} \tag{4}$$

Esempio: Se un agente dovesse trovare x almeno buono quanto y, allora la funzione di utilità $u(x)$ deve essere almeno alta quanto $u(y)$. Cioè l'agente è come se stesse **massimizzando** il valore di $u(var)$.

Rappresentazione ordinale Teorema 1: Sia $\mathcal{X}$ un insieme finito di alternative e sia $\succcurlyeq$ una relazione di preferenza su $\mathcal{X}$. Allora una preferenza può essere rappresentata come una funzione di utilità $u : \mathcal{X} \to \mathbb{R}$ se e solo se è **completa** e **transitiva**. In più, se $f : \mathcal{R} \to \mathcal{R}$ è una funzione monotona crescente, allora $f \circ u$ rappresenta la stessa preferenza di $u \succcurlyeq$

Nota: dall'ultimo statement, l'ordine ha rilevanza.

Per essere valido ci sono 2 condizioni necessarie:

- Transitività: Cioè, dato $\mathcal{X} = \{a, b, c\}$, suponiamo che $a \succ b \succ c \succ a \implies u(a) > u(b) > u(c) > u(a)$. Questo sarebbe **assurdo**.
- Completezza: Se abbiamo preferenze incomplete, allora al massimo possiamo costruire un ordine per un sottoinsieme di $\mathcal{X}$.

Dimostrazione

La transitività e la completezza sono necessarie e sufficienti. Supponiamo di avere l'insieme $X = \{X_1, \ldots, X_n\}$. Possiamo suddividere gli elementi di X in k classi di indifferenza $C_1, \ldots, C_k$ tali che $C_1 \succ C_2 \succ \ldots \succ C_k$. In questo modo, possiamo definire la funzione di utilità u in modo che:

$$\begin{gathered} u(x) = k \quad \forall x \in C_1, \\ u(x) = k - 1 \quad \forall x \in C_2, \\ \ldots \\ u(x) = 1 \quad \forall x \in C_k. \end{gathered}$$

In questo contesto, $\succ$ rappresenta la relazione di preferenza.

4.2 Le lotterie

Una lotteria è una tupla $\mathcal{L} = (p_1, x_1; p_2, x_2 \ldots, p_n, x_n)$.

- Con prezzo monetario $x_1, x_2, \ldots, x_n \in X \subseteq R$.
- Distribuzione di probabilità $(p_1, p_2, \ldots, p_n)$.

Quindi con $\mathcal{L}$ viene definito l'insieme delle lotterie semplici.