Università degli studi di Milano-Bicocca

Decision Models

Final Project

Fase di attacco di una squadra di calcio durante un match: Approximated Dynamic Programming per il Decision Making

---

Autore:
Mauro Samarelli – 834196 – m.samarelli@campus.unimib.it

Giugno 21,
2019

Abstract

Questo progetto ha l'obiettivo di servirsi dell'Approximated Dynamic Programming per studiare lo svolgimento delle azioni di attacco di una squadra di calcio ed analizzare come varia il loro sviluppo nel corso di un match. Lo scopo di questo studio è quello di analizzare il grado con cui, nel corso di un match, la squadra in oggetto ha utilizzato (consapevolmente o non) l'apprendimento per rinforzo, ovvero si è servita delle informazioni ottenute dagli esiti di azioni precendenti per poter migliorare lo sviluppo di quelle future e aumentare la possibilità di segnare un gol. Tale concetto si basa sull'idea di pensare una squadra di calcio (composta da 11 calciatori) come un sistema unico, il quale durante un match, assume un comportamento paragonabile a quello di un agente che interagisce con l'ambiente in cui si trova e impara ad agire secondo i reward e/o punishment ottenuti da queste interazioni. Durante un match, il sistema squadra mette in atto delle azioni di attacco complesse, ma divisibili in singoli gesti tecnici e/o tattici, con lo scopo di portare la palla il più vicino possibile alla porta avversaria ed effettuare un tiro per segnare un gol. Da ogni singola interazione, ovviamente soggetta alla presenza di una squadra avversaria, il sistema squadra riceve un riscontro (reward se l'interazione ha permesso di avvicinarsi all'obiettivo, punishment in caso contrario), e quindi in modo più o meno esplicito ne fa uso in futuro attuando la politica di apprendimento per rinforzo (Reinforcement Learning). L'Approximated Dynamic Programming permette di gestire l'apprendimento per rinforzo in cui il reward/punishment di un'azione non è determinato a priori, ma si basa sul valore atteso (expected value). Il valore atteso del reward di un'azione elementare, all'interno di un'azione di attacco di una squadra, viene osservato con una diretta interazione con l'ambiente (la squadra fa un'azione, osserva il reward/punishment, aggiorna il suo apprendimento e quando si trova in uno stato uguale è in possesso di informazioni aggiornate sul valore atteso del reward/punishment che otterrebbe rifacendo quella azione). Ciò sta alla base della scelta di dividere il match in due parti: la prima, dall'inizio del match al 15esimo minuto, usata come fase di "Exploration" dell'algoritmo (ovvero la squadra mette in atto la politica "act-to-observe" perchè non ha informazioni sufficienti sul valore atteso del reward di un'azione → in gergo calcistico "studio dell'avversario"); la seconda, dal 15esimo minuto alla fine del match, usata come fase di "Exploitation" dell'algoritmo (ovvero la squadra mette in atto la politica "act-observe-improve to

get target” per raggiungere l’obiettivo di segnare un gol servendosi delle informazioni che ha a disposizione e aggiornandole dopo ogni azione di attacco). Prima di procedere con la presentazione delle tecniche utilizzate per questo studio, è doveroso premettere che questo progetto fa uso di una visione semplificata dell’ambiente campo da calcio durante un match, in quanto tralascia molte variabili che possono essere determinanti per l’esito del match stesso o di un’azione (in primis la politica di gioco in difesa dell’avversario, la quale comunque potrebbe essere controllata inserendo nello studio informazioni relative al comportamento attuato in match precedenti contro avversari che difendono in modo simile, ma anche altri aspetti non individuabili dalla semplice osservazione del movimento della palla verso il target, come ad esempio la condizione fisico-atetica e psicologica dei calciatori, oppure fattori esterni come le condizioni metereologiche, condizioni del campo da gioco, influenza del pubblico o altro).

1 Introduzione

L’oggetto di studio di questo progetto è la fase di attacco della Nazionale Spagnola di Calcio messa in atto durante il match Spagna-Svezia, valido per le qualificazioni agli Europei di Calcio 2020, disputato il 10 Giugno 2019 allo Stadio Santiago Bernabeu di Madrid e terminato con il punteggio di 3-0 in favore della Spagna. La scelta di questo evento non è casuale, in quanto vede contrapposte su un campo da calcio due squadre con caratteristiche di gioco completamente opposte e che si prestano bene alla valutazione dell’apprendimento per rinforzo da parte della Spagna per lo svolgimento delle azioni di attacco. La Spagna, infatti, è nota per il suo stile di gioco offensivo che fa uso di una costante ricerca del gol tramite azioni di attacco composte da una serie numerosa di passaggi, mentre la Svezia è nota per il suo stile di gioco estremamente difensivo che ha lo scopo di impedire agli avversari di segnare. La fase preliminare alla realizzazione e valutazione dell’algoritmo di apprendimento per rinforzo, consiste in due operazioni:

1. dividere concettualmente il campo da gioco in 18 zone con lo scopo di associare la situazione di avere il possesso della palla nella zona i-esima allo stato i-esimo tra quelli definiti per l’algoritmo (vedi Paragrafo 3 – Approccio Metodologico)

Figura 1: divisione concettuale in zone del campo da gioco

2. tramite un software di “video-analysis” effettuare il tagging delle azioni elementari in cui è scomponibile un’azione di attacco sviluppata dalla Spagna (sono state escluse le palle inattive e i loro sviluppi); per convenzione semplificatrice, una singola azione è riferita al movimento della palla da una zona ad un’altra delle 18 in cui è stato diviso il campo, avvenuto tramite una specifica modalità tra una lista di possibili modalità considerate. Ogni singola azione ha un outcome: positivo, se il possesso della palla è stato mantenuto; negativo, se il possesso della palla è stato perso (vedi Paragrafo 2 – Dataset). Terminato il tagging, è possibile ricavare un dataset grezzo relativo alle informazioni sulle azioni elementari “taggate”, esportandolo dal software di “video-analysis” utilizzato.

2 Datasets

Per descrivere efficientemente la natura del dataset ottenuto dalla fase preliminare di “tagging”, è necessario specificare i vincoli per la classificazione della singola azione:

1. ad ogni azione deve essere associata solo una zona di campo di partenza (da 1 a 18);
2. ad ogni azione deve essere associata solo una zona di campo di arrivo (da 1 a 18 o il target → ricerca della porta avversaria);
3. ad ogni azione deve essere associata solo una modalità tra quelle della lista presente nel file “actions_weigth.csv”;
4. ad ogni azione deve essere associato solo un outcome (esito) tra “positivo” e “negativo”.

Il dataset “spain-sweden.csv” che raccoglie i dati relativi alle azioni elementari che compongono un’azione di attacco della Spagna durante il match ha le seguenti caratteristiche:

● 779 osservazioni o unità statistiche (ognuna di queste descrive una singola azione elementare avvenuta durante un’azione di attacco)

● 56 colonne o attributi, di seguito descritte:
- Name (categoriale): id progressivo dell’azione elementare
- Time, Start, Stop (numerici): riferimenti temporali dell’azione elementare
- Team, Player (categoriali): riferimenti squadra e calciatore che effettua l’azione elementare (NB: in questa analisi sono stati tralasciati i tag sui calciatori)
- From1 – From18 (binarie): indicano da quale zona di campo è partita l’azione elementare
- To1 – To18, ToTarget (binarie): indicano verso quale zona di campo è diretta l’azione elementare
- 11 attributi binari sulla modalità dell’azione elementare (vedi in seguito descrizione file “actions_weight.csv”)
- Positivo, Negativo (binarie): indicano l’outcome (esito) dell’azione elementare

Oltre al dataset principale (file “spain-sweden.csv”) precedentemente descritto,

vi sono anche altre due strutture dati di supporto:

1. file “actions_weigth.csv” contenente i pesi (quindi il valore) teorici e modificabili secondo le esigenze, assegnati a ciascuna modalità di azione con la quale si parte da una zona i e si arriva a una zona j nel campo (i può essere anche uguale a j).

Le modalità considerate sono: passaggio in orizzontale, passaggio in avanti, retropassaggio, passaggio chiave, conduzione della palla, cambio di fascia, assist, conclusione in porta, palla gol creata, rigore conquistato e gol fatto. Essendo il calcio uno sport dove l’evento di raggiunta del target (segnare un gol) è molto raro rispetto alla quantità di azioni elementari eseguite, per le ultime quattro modalità non è stato previsto un peso per l’outcome negativo in quanto si vuole sempre premiare la ricerca del target a prescindere dall’esito (discorso diverso riguarda altri sport come basket, volley o rugby).

2. file “checkpoint.csv” contenente l’id progressivo dell’ultima azione elementare compresa in un determinato periodo di tempo del match (vedi Paragrafo 3.2 –Scelta dei checkpoint)

3 Approccio Metodologico

In questa sezione viene presentata e giustificata l’implementazione dell’apprendimento per rinforzo sinora descritto.
Prima di tutto bisogna specificare che l’obiettivo pratico è quello di ottenere un insieme di possibili path che collegano le zone del campo da calcio, associate a sequenze di decision making, potenzialmente ottimali per lo sviluppo di un’azione di attacco che possa aumentare la probabilità di segnare un gol.
L’informational setting di tale ambiente è di tipo stocastico, in quanto bisogna considerare transizioni dinamiche di tipo probabilistico da una zona ad un’altra e valutarne i reward sotto forma di expected value. Si tratta quindi di adottare una strategia risolutiva ad un problema di natura Markov Decision Process (MDP).

3.1 Definizione Ambiente

Insieme degli stati
$S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, TARGET\}$

Durante un match, il sistema squadra è nello stato $s \in S$ in ogni momento in cui ha il possesso della palla nella zona del campo ad esso associata.
La sequenza di transizioni dinamiche di tipo probabilistico può iniziare e terminare in qualsiasi stato (possesso della palla recuperato in una zona che dà inizio all’azione di attacco oppure possesso della palla perso in una zona che termina l’azione di attacco).
Solo lo stato TARGET è uno stato in cui l’azione può soltanto terminare ed è riferito alla ricerca della porta avversaria (non necessariamente con raggiunta del bersaglio)

Insieme delle azioni
A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, TARGET}

Ipotizzando che il sistema squadra sia in uno dei possibili stati
s ∈ S - {TARGET}, esso effettua un'azione elementare, ovvero entra nel processo di decision making per far transitare la palla da una zona ad un'altra del campo. Ogni a ∈ A rappresenta quindi il passaggio dallo stato s ∈ S - {TARGET} allo stato s' ∈ S, con s' coincidente con a. La natura probabilistica deriva dal fatto che, essendo nello stato s vi sono probabilità diverse di effettuare un'azione a per raggiungere uno degli stati s' (≡ a). Ogni azione a, indipendentemente dallo stato di partenza e lo stato di arrivo, viene classificata e pesata secondo una specifica modalità con cui è stata eseguita (file "actions_weight.csv") e secondo il suo esito (positivo/negativo). Ciò permette di definire in modo stocastico il valore delle transizioni dinamiche da ogni stato s ∈ S - {TARGET} ad ogni stato s' ∈ S tramite l'expected reward.
Il numero delle transizioni dinamiche di tipo probabilistico è quindi:
| S - {TARGET} | x | S | = 18 x 19 = 342

3.2 Scelta dei checkpoint

Lo step di apprendimento del sistema squadra è definito come l'intervallo di tempo del match compreso tra due checkpoint, dove per checkpoint si intende uno specifico minuto del match. Nel file "checkpoint.csv" sono stati definiti 7 checkpoint, oltre a quello implicito del minuto 0 coincidente con l'inizio del match. Per il primo tempo sono stati fissati 3 checkpoint a distanza di 15 minuti uno dall'altro (15', 30', 45'). Per il secondo tempo sono stati fissati altri 4 checkpoint: al minuto 62', coincidente con la fine dell'azione di attacco precedente a quella correlata alla conquista del rigore che ha permesso alla Spagna di segnare il primo gol, al minuto 75', al minuto 83', coincidente con la fine dell'azione di attacco precedente a quella correlata alla conquista del rigore del secondo gol e, infine, al minuto 87' coincidente con la fine "virtuale" del match, decretata dal terzo gol della Spagna. In generale il checkpoint dovrebbe essere definito in base a un determinato lasso di tempo trascorso, la segnatura di un gol o un evento che potenzialmente potrebbe influire sul match, come un'espulsione, una sostituzione o un cambio di modulo del sistema squadra e/o avversari.

3.3 Fase di Exploration

La fase di "Exploration" riguarda il periodo del match che va dal suo inizio al 15esimo minuto (scelta progettuale) e ha lo scopo di utilizzare l'algoritmo di Value Iteration del Reinforcement Learning per ottenere i valori della funzione di stato per ogni zona del campo da calcio, inizializzati a valore nullo all'inizio del match. E' necessario specificare che questa fase rappresenta solo il primo step dell'algoritmo, il quale ha un numero di step pari al numero di checkpoint inseriti

nel file “checkpoint.csv”. Ad ogni checkpoint, l’algoritmo aggiorna i valori della funzione di stato per ogni zona del campo, terminando in corrispondenza della fine del match. Il primo checkpoint considerato è il 15esimo minuto, corrispondente quindi al termine della fase di “Exploration”.

VALUE ITERATION
Start of match: $V(s) \leftarrow 0 \ \forall \ s \in S$
checkpoint ← 1
Repeat
  for all $s \in S - \{TARGET\}$
    for all $a \in A$
      $Q(s, a) \leftarrow E[r \mid s, a] + \gamma \sum_{s' \in S} P(s' \mid s, a) * V(s')$
    $V(s) \leftarrow \max_a Q(s, a)$
  $V(TARGET) \leftarrow \max V(s)$
  checkpoint ← checkpoint + 1
Until Time(checkpoint) = End of match

NB: V(S) alla fine dello step durante il quale checkpoint passa da 0 a 1 rappresenta il vettore dei valori della funzione di stato al termine della fase di “Exploration”

$$Q(s, a) \leftarrow E[r \mid s, a] + \gamma \sum_{s' \in S} P(s' \mid s, a) * V(s')$$

E[r | s, a]: expected reward associato ad effettuare l’azione a nello stato s, ovvero rappresenta il valore atteso del reward che ottiene il sistema squadra quando effettua l’azione elementare a per far transitare la palla dallo stato s allo stato s’ (≡ a)

Esempio: supponendo che al checkpoint x il sistema squadra, quando si è trovato nello stato 5, ha effettuato 3 azioni elementari per far transitare la palla nello stato 10 (classificate nelle seguenti modalità: 2 passaggi chiave positivi con singolo peso pari a 2 e 1 passaggio chiave negativo con singolo peso pari a -0.5), l’expected reward da s=5 con a=10 (≡ s’) è:

E[r | 5, 10] = score_medio_azioni_positive * prob_azione_positiva + score_medio_azioni_negative * prob_azione_negativa =
= 2 * (2/3) + (-0.5) * (1/3) = 1.32 – 0.165 = 1.155

γ : fattore di sconto compreso tra 0 e 1, il cui valore dipende dalla natura del problema circa l’orizzonte finito o infinito per la convergenza dell’algoritmo (in questo progetto si è scelto $\gamma = 1$)

P(s’ | s, a): probabilità della transizione dinamica dallo stato s allo stato s’ effettuando l’azione a. Considerato che, nel caso di questo algoritmo,

s’ ≡ a, la natura probabilistica non riguarda l’accuratezza con la quale si raggiunge lo stato successivo s’, bensì la frequenza con la quale si passa dallo stato s allo stato s’ in rapporto alla frequenza delle transizioni complessive partite da s.

Esempio: supponiamo che al checkpoint x il sistema squadra si è trovato 10 volte nello stato s=5 e ha effettuato 4 volte un’azione a per raggiungere lo stato s’=10, 3 volte un azione a per raggiungere lo stato s’=8 e 3 volte un azione a per raggiungere lo stato s’=9 (indipendentemente da modalità e outcome di a):

P(10 | 5, 10) = freq(5 → 10) / freq(5 → *) = 4/10 = 0.4
P(8 | 5, 8) = freq(5 → 8) / freq(5 → *) = 3/10 = 0.3
P(9 | 5, 9) = freq(5 → 9) / freq(5 → *) = 3/10 = 0.3
NB: il simbolo “*” indica un qualsiasi stato di S

V(S’) : indica il valore della funzione di stato nello stato s’ di arrivo (quindi valutata anche nello stato TARGET posto uguale al valore della funzione di stato più alto valutata per gli altri stati). Rappresenta “quanto ha valore avere il possesso della palla nella zona di campo associata allo stato s” → parametro quantitativo statico

Q(s, a) : indica il valore della funzione di azione, dato lo stato di partenza s e l’azione a, la quale consente di effettuare una transizione dinamica verso lo stato s’ (≡ a). Rappresenta “quanto ha valore avere il possesso della palla nella zona di campo associata allo stato s e decidere di far transitare la palla nella zona di campo associata allo stato s’ tramite un’azione a” → parametro quantitativo-qualitativo dinamico

Un’azione a che permette di transitare dallo stato s allo stato s’ è definita come una policy π (s). Si definisce policy ottimale π * (s) per ogni s ∈ S – {TARGET}, quella policy che definisce un’azione a in grado di transitare dallo stato s allo stato s’ (≡ a), massimizzando la funzione di azione Q(s, a).

$\pi^* : Q_{\pi^*}(s, a) >= Q_{\pi}(s, a)$ for all s, all a and all π

L’algoritmo Value Iteration, per ogni checkpoint fissato a priori, assegna ad ogni stato s un valore V(s) che coincide con il valore massimo della funzione di azione considerata per gli elementi che hanno lo stato s come stato di partenza (utilizzando quindi la policy ottimale π * (s))

$V_{\pi^*}(s) = \max_{a \in A} Q_{\pi^*}(s, a)$ for all s and all a

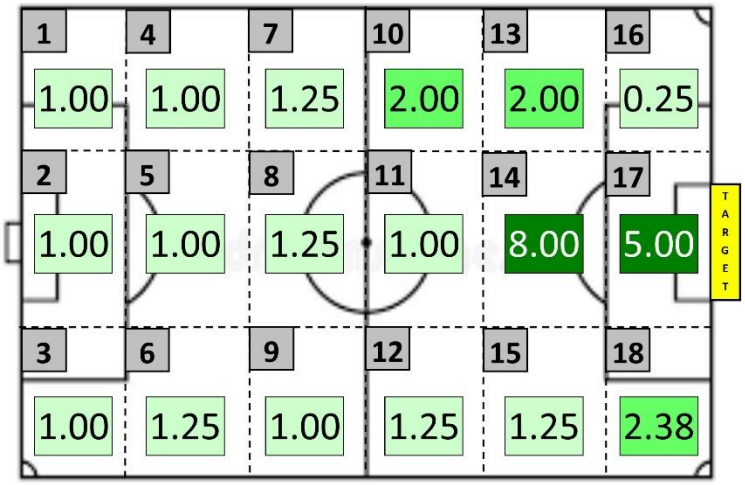


*Figura 2: Valori della funzione di stato **V π * (s)** per ogni zona del campo in riferimento alla fase di attacco della Spagna al termine della fase di Exploration (15esimo minuto del match Spagna-Svezia)*

Definito uno stato di partenza s ∈ S – {TARGET}, è necessario specificare che l'unione delle policy ottimali π * (s) per ogni s ∈ S – {TARGET} non necessariamente compone un path che raggiunge il target.
Da ciò consegue che ottenere un path ottimale per arrivare al target, significa costruire un percorso che ha come stato di partenza lo stato s definito, come stato di arrivo lo stato "target" e una sequenza di transizioni dinamiche che non necessariamente devono essere in accordo con le singole policy ottimali π * (s) di ogni stato.
Il path ottimale, in ogni caso, sarà costruito partendo dal target e selezionando a ritroso gli stati che massimizzano il valore della funzione di azione ad ogni transizione, vietando di considerare una transizione in cui lo stato di partenza è stato già considerato in precedenza. Tradotto in termini calcistici, il path ottimale definisce la sequenza di azioni da mettere in atto per muovere la palla da una zona ad un'altra del campo, senza tornare in una zona già considerata, con l'obiettivo di guadagnare il massimo vantaggio da ogni movimento ed arrivare al target. Tale path ottimale è definibile per ogni zona del campo, quindi per ogni stato s ∈ S – {TARGET}, e alla fine di ogni checkpoint.

La presenza di un'unione di policy ottimali π * (s) che non raggiunge il target, tornando durante il percorso in uno stato già considerato, potrebbe essere un campanello d'allarme sulle difficoltà del sistema squadra di sviluppare azioni di attacco efficienti. Analizzando la dinamica delle transizioni si può riuscire a scoprire in quali zone del campo è più evidente questo problema.

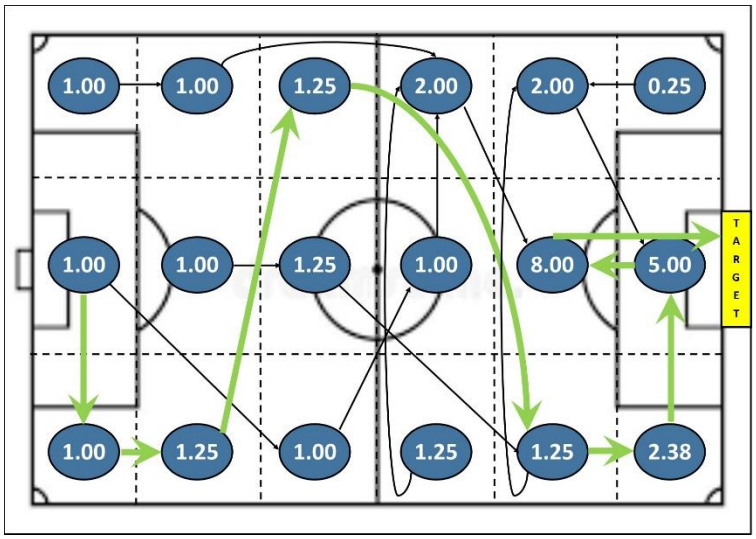
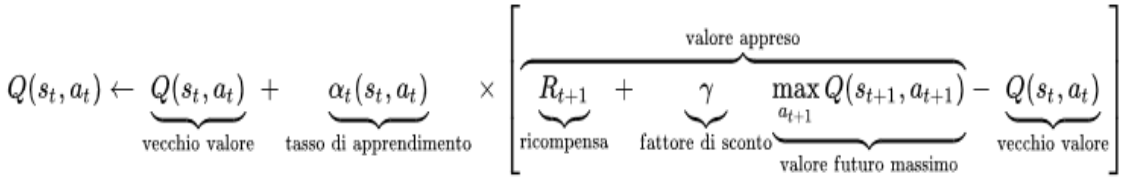


*Figura 3: Valori della funzione di azione **Q π * (s, a)** per ogni zona del campo in riferimento alla fase di attacco della Spagna al termine della fase di Exploration (15esimo minuto del match Spagna-Svezia)*
Ipotizzando una rimessa del portiere, il path ottimale è rappresentato dalle frecce di colore verde

Il limite della matrice Q ottenuta ad ogni iterazione dell'algoritmo Value Iteration (quindi per ogni checkpoint) è di rappresentare una "fotografia" della situazione relativa ad ogni checkpoint senza esprimere il grado di apprendimento avuto dal sistema squadra tra un checkpoint e un altro, perciò l'utilità della fase di "Exploration" consiste essenzialmente nell'inizializzare i valori della matrice Q che sarà aggiornata per iterazioni di apprendimento durante la fase di "Exploitation".

3.4 Fase di Exploitation

Fissato un checkpoint, per ricavare una matrice Q significativa del grado di apprendimento per rinforzo ottenuto rispetto al checkpoint precedente, si è scelto di implementare l'algoritmo Q-Learning. Si tratta di un algoritmo di rinforzo off-policy e il suo vantaggio più rilevante è l'abilità di comparare l'utilità aspettata delle azioni disponibili (expected reward) senza richiedere un modello dell'ambiente. Come spiegato precedentemente, la matrice Q viene inizializzata con quella ottenuta alla fine della fase di "Exploration". Successivamente ogni elemento della matrice Q viene aggiornato ad ogni checkpoint con la seguente formula:



α (s, a) : indica il tasso di apprendimento che, per lo scopo di questo progetto, è stato reso variabile per ogni coppia (s, a). Rappresenta “quanto il sistema squadra ha potuto imparare trovandosi nello stato s ed effettuando l’azione a per transitare in s’ (≡ a)”, quindi viene calcolato come probabilità di transizione nel periodo del match compreso tra due checkpoint:

α (s, a) = freq(s → s’(≡ a)) / freq(s → *)
NB: il simbolo “*” indica qualsiasi stato di S

E [r_{t+1} | s_{t+1}, a_{t+1}] : valore atteso del reward ottenuto dal sistema squadra, effettuando un’azione a nello stato s per transitare nello stato s’ (≡ a), riferito al periodo del match compreso tra il checkpoint precedente (tempo t) e il checkpoint raggiunto (tempo t+1), diversamente dal valore atteso del reward calcolato per ogni step del Value Iteration (vedi Paragrafo 3.3 – Exploration), il quale era riferito al periodo del match compreso dal suo inizio al checkpoint raggiunto.

γ : fattore di sconto posto a 1 (scelta progettuale)

$\max_{a(t+1)} Q(s_{t+1}, a_{t+1})$: valore della funzione di azione per ogni stato s massimizzata dall’azione a, considerando la matrice Q ottenuta per il periodo di tempo del match compreso tra il checkpoint precedente (tempo t) e il checkpoint raggiunto (tempo t+1), diversamente dalla matrice Q ottenuta ad ogni step del Value Iteration (vedi Paragrafo 3.3 – Exploration), la quale era riferita al periodo del match compreso dal suo inizio al checkpoint raggiunto.

L’algoritmo Q_Learning così descritto è eseguito nel seguente modo:

1. inizializzo Q [tempo=t] con Q [Exploration] ottenuta al checkpoint 1
Per ogni checkpoint CK successivo a 1 (fino a fine match):
   2. calcolo la matrice Q [tempo=t+1] riferita al periodo [CK-1; CK]
   3. ricavo il valore massimo Q [tempo=t+1] per ogni stato s
   4. aggiorno ogni elemento di Q [tempo=t] per ogni stato s

NB: I valori della funzione di stato necessari al calcolo della Q[tempo=t+1] fra due checkpoint [CK-1; CK] sono quelli riferiti al checkpoint CK-1, precedentemente ottenuti tramite l’algoritmo Value Iteration (vedi Paragrafo 3.3 – Exploration).

4 Results and Evaluation

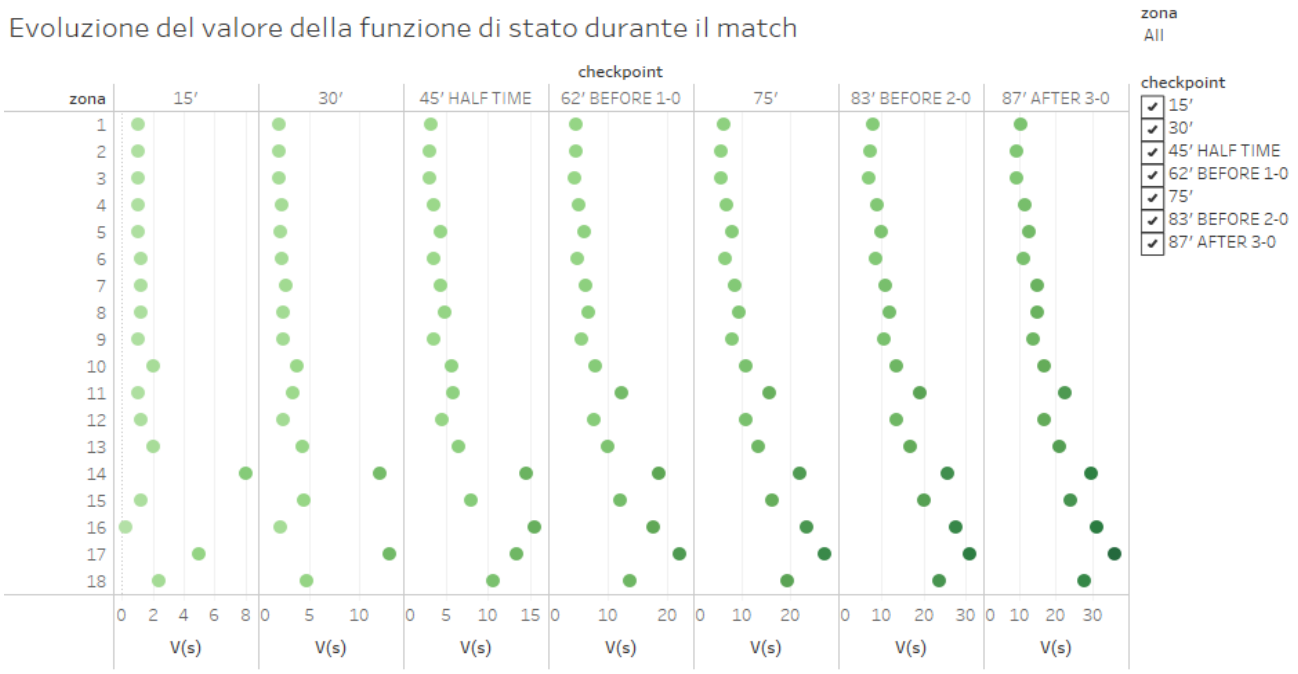


Figura 4: Evoluzione di V(s) durante il match
La dashboard interattiva, sviluppata con Tableau, è disponibile al link https://public.tableau.com/views/EvoluzioneVs/Dashboard1?:embed=y&:display_count=yes&publish=yes&:origin=viz_share_link

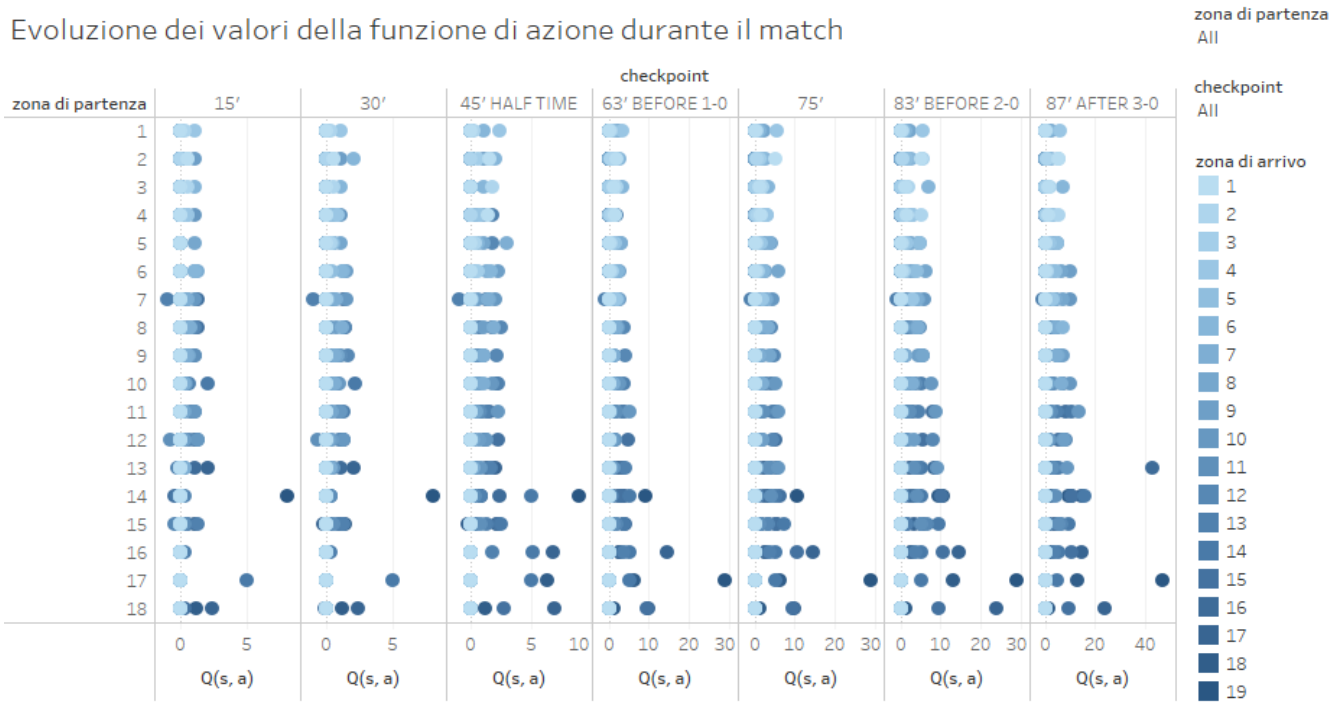


Figura 5: Evoluzione di Q(s, a) durante il match
La dashboard interattiva, sviluppata con Tableau, è disponibile al link
https://public.tableau.com/views/EvoluzioneQsa/Dashboard1?:embed=y&:display_count=yes&publish=yes&:origin=viz_share_link

5 Discussione

Dato il vasto insieme di considerazioni circa l'analisi dell'apprendimento per rinforzo di cui può far uso il sistema squadra durante un match, determinato dal numero di zone del campo (stato), dal numero di transizioni (stato x azione) e dal numero di checkpoint considerati, viene riportata la discussione dei risultati ottenuti per un'analisi specifica. Scegliendo di voler analizzare come è variato l'utilizzo e il vantaggio ottenuto dal gioco sulle fasce, in particolare se è stata utilizzata in modo migliore la zona 16 (zona laterale a sinistra più vicina all'area di rigore avversaria) o la zona 18 (zona laterale a destra più vicina all'area di rigore avversaria), di seguito è riportato il risultato delle dashboard interattiva sviluppata con Tableau:

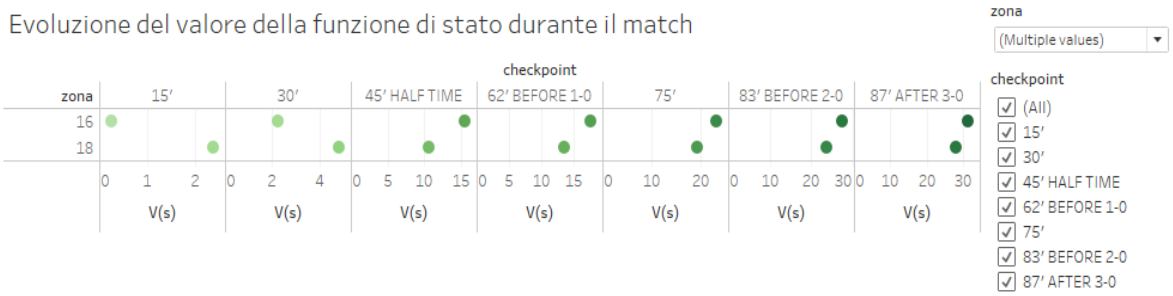


Figura 6: Confronto andamento di V(s) durante il match per le zone 16 e 18

Guardando l'evoluzione di V(s), si può notare che fino al 30' del primo tempo il valore riferito ad avere il possesso della palla nella zona 18 (a destra dell'area di rigore avversaria) è maggiore rispetto a quello riferito ad avere il possesso della palla nella zona 16 (a sinistra dell'area di rigore avversaria). Tale considerazione cambia in maniera opposta nel corso del periodo del match compreso tra il 30' e la fine del primo tempo e viene confermata per tutto il secondo tempo.

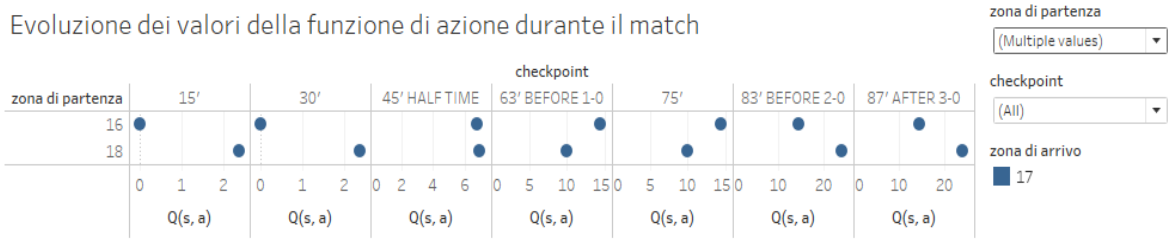


Figura 7: Confronto andamento di Q(s, a) durante il match per le zone 16 e 18

Guardando l'evoluzione di Q(s, a), si può notare che fino al 30' del primo tempo il valore riferito ad avere il possesso della palla nella zona 18 (a destra dell'area di rigore avversaria) e decidere di raggiungere la zona 17 (area di

rigore avversaria) è maggiore rispetto a quello riferito a raggiungere la stessa zona avendo il possesso della palla nella zona 16 (a sinistra dell'area di rigore avversaria). Tale considerazione cambia nel corso del periodo del match compreso tra il 30' e la fine del primo tempo, alla fine del quale i valori sono sostanzialmente uguali. Nel secondo tempo, fino al 75' minuto vale la considerazione opposta e dal 75' fino al gol del 3-0 torna a valere questa considerazione.

NB: la conquista del primo rigore che ha sbloccato la partita (minuto 63' → valori V(s) e Q(s, a) aggiornati al checkpoint "Before 1-0") è nata da un fallo di mano commesso da un difensore svedese nella zona 16 che ha intercettato un cross (azione classificata come tentativo di assist positivo), effettuato da un calciatore spagnolo e proveniente dalla zona 16 verso la zona 17. Lascio al lettore la considerazione sulla classificazione di questo evento come il "fatidico episodio che sblocca la partita" tanto citato nell'analisi calcistica di un match, oppure come una consegenza del processo di apprendimento per rinforzo svolto dalla Spagna fino a quel momento del match per sviluppare azioni di attacco efficaci.

6 Conclusioni

Supponendo di eseguire l'algoritmo proposto per valutare l'apprendimento per rinforzo della propria squadra e/o quella avversaria riguardante gli ultimi match disputati, lo staff tecnico di una squadra di calcio avrebbe a disposizione un patrimonio di informazioni aggiuntivo davvero significativo e di alto valore per la preparazione di un match.

I limiti di questo progetto consistono nel lavoro molto oneroso della fase di tagging delle azioni elementari effettuate durante un match, a cui si aggiunge la trasmissione del valore dei contenuti allo staff tecnico, la quale deve avvenire con un linguaggio semplice, comprensibile e adatto al contesto calcistico senza entrare nei particolari dell'implementazione (questa tipologia di analisi verrebbe immediatamente cestinata in caso contrario). Volendo apportare accorgimenti tattici durante un match basandosi sul valore delle informazioni ricavate dal metodo di analisi presentato, sarebbe necessaria la progettazione di una complessa architettura in grado di fornire queste informazioni in tempo reale.

References

[1] Marco Wiering, Rafal Salustowicz, Jurgen Schmidhuber, “Reinforcement Learning Soccer Teams with Incomplete World Models”
https://pdfs.semanticscholar.org/1614/55516e8f8cfb1e8fe7a5b7ebe32ca4980992.pdf

[2] Dapeng Zhang, “Action Selection and Action Control for playing Table Soccer Using Markov Decision Process”
http://www2.informatik.uni-freiburg.de/~ki/papers/zhang-master-thesis-05.pdf

[3] Marcus Post, Oliver Junge, “Exploiting Symmetries in Two Player Zero-Sum Markov Games with an Application to Robot Soccer”
https://pdfs.semanticscholar.org/cd62/aaf3a9f7fcd8d4271a06b19814c13e8a361a.pdf

[4] Gabriel Damour, Philip Lang, “Modelling Football as a Markov Process”
https://www.diva-portal.org/smash/get/diva2:828101/FULLTEXT01.pdf