Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in
Scienze Statistiche

Metodi per la stima dinamica del ranking dei giocatori NBA

Relatore Prof. Nicola Sartori
Dipartimento di Scienze Statistiche

Laureando Pietro Gioia
Matricola 2056748

Anno Accademico 2022/2023

Ai miei genitori

Indice

Introduzione **1**

1 Il *dataset* **3**
1.1 Fonte di dati . . . 3
1.2 Le variabili . . . 4
1.3 Trattamento iniziale dei dati . . . 6
1.3.1 Evento e sotto-evento . . . 7
1.3.2 Creazione della variabile risposta . . . 9
1.4 Indici statistici esistenti . . . 10

2 Approccio statico **13**
2.1 Modello *probit* . . . 13
2.1.1 Modello *stepwise* . . . 16
2.1.2 Inserimento effetti fissi . . . 18
2.2 Modello *probit* con correzione di Firth . . . 21
2.2.1 Costruzione del *ranking* . . . 23
2.2.2 *Ranking* corretto . . . 24
2.2.3 Lisciamento a media mobile . . . 25
2.3 Classifica finale . . . 28
2.3.1 Classifica per ruolo . . . 30
2.3.2 Classifica per squadra . . . 32
2.3.3 Regular Season e Playoff . . . 34
2.4 Valutazioni . . . 35

3 Approccio bayesiano **37**
3.1 Inferenza bayesiana . . . 37
3.2 Approssimazione normale . . . 39
3.2.1 Aggiornamento normale-normale . . . 41
3.2.2 Modifica tramite *power prior* . . . 43
3.2.3 Considerazioni e risultati . . . 47
3.3 Distribuzione *SUN* . . . 49
3.3.1 *SUN* coniugata . . . 50
3.3.2 Funzione di ripartizione multivariata normale . . . 51
3.3.3 Problemi computazionali . . . 52

3.4 Algoritmo EP 54
3.4.1 EP e distribuzione a posteriori 55
3.4.2 Classifica EP e normale-normale 60
3.5 Valutazioni 62

Conclusioni **62**

Appendice Codice R utilizzato **65**

Bibliografia **77**

Introduzione

La statistica è una disciplina fondamentale nel mondo moderno, il cui ruolo è cresciuto in modo straordinario grazie alla rapida diffusione di enormi moli di dati e all'aumento della potenza di calcolo. Questa trasformazione ha radicalmente influenzato la raccolta, l'analisi e l'interpretazione delle informazioni in svariati settori, dall'industria alla scienza, dalla letteratura allo sport. Nel mondo sportivo, come nel calcio, nella pallacanestro o nel tennis, l'impiego di dati e analisi statistiche per ottimizzare le prestazioni degli atleti è diventato una professione a sé stante, ridefinendo la percezione e l'approccio a tali discipline. Un esempio emblematico è il *basket*, un contesto particolarmente adatto all'analisi statistica, data la sua ampia diffusione e la grande quantità di dati disponibile. In questo contesto, l'uso di indicatori basati su dati, quali la percentuale di successo nei tiri a canestro, i rimbalzi catturati, i falli commessi e molti altri, riveste un ruolo centrale nell'analisi delle prestazioni dei giocatori e nella creazione di eventuali classifiche.

Il presente elaborato si pone l'obiettivo di approfondire quest'ultimo aspetto, ossia aggiornare la classifica dei migliori giocatori della National Basketball Association (NBA) nella stagione 2018/19, supponendo di essere in un caso di *data stream*, ossia di ricevere nuovi dati di giornata in giornata e di aggiornare dinamicamente la classifica. In questi casi, una delle principali sfide è quella di aggiornare le stime ottenute fino al momento corrente utilizzando solamente i dati in arrivo, velocizzando dunque le procedure ed evitando di dover ripetere le analisi includendo dati passati. Le prestazioni dei giocatori verranno valutate a partire dalle azioni degli incontri a cui hanno partecipato attraverso un indice di pericolosità costruito *ad hoc* per la presente tesi.

L'elaborato si organizza come segue: nel Capitolo 1 viene presentato il *dataset* e tutte le problematiche affrontate in fase di preparazione dei dati prima della costruzione della matrice del disegno finale. Viene inoltre discussa la creazione della variabile risposta a partire dalle azioni di gioco disponibili. Prima di procedere con i metodi utilizzati,

vengono inoltre presentate alcune classifiche dei giocatori provenienti da fonti esterne, in modo tale da avere un termine di paragone.

Nel Capitolo 2 viene introdotto il modello di regressione binaria con *link probit*, il quale presenta tuttavia alcuni problemi di stima dovuti alla quasi perfetta separazione tra le variabili. Questo perché, in questa fase, si considerano le giornate come indipendenti e si stimano dei modelli *probit* separati sui soli dati di ogni singola giornata di gioco, con la conseguenza di dover utilizzare un limitato numero di osservazioni per adattare il modello. Per ovviare a questo problema viene introdotta la correzione per la distorsione proposta da Firth (1993), con la quale è possibile ottenere delle stime più affidabili dei parametri. Le previsioni fornite dal modello verranno poi modificate in modo da includere una dipendenza temporale. Vengono inoltre eseguite delle analisi dei risultati stratificate per ruolo, squadra e periodo di campionato.

Nel Capitolo 3 vengono approfonditi degli approcci bayesiani, con lo scopo di includere informazione passata nell'aggiornamento dei parametri, evitando così di dover stimare modelli sui soli dati relativi alla giornata di interesse. Un primo approccio prevede di aggiornare, con l'arrivo dei dati, i parametri di una distribuzione a priori normale e di utilizzare la nuova distribuzione a posteriori ottenuta come priori per la giornata successiva. Successivamente, vengono introdotti dei coefficienti, tramite la tecnica della *power prior*, in modo tale da pesare le vecchie partite proporzionalmente alla distanza con la partita corrente. Un secondo approccio prevede, invece, di utilizzare la distribuzione SUN (Arellano-Valle & Azzalini, 2006) per l'aggiornamento dei parametri del modello, essendo quest'ultima la distribuzione coniugata del modello *probit* (Durante, 2019). Infine, viene presentata una versione efficiente dell'algoritmo EP per modelli *probit* (Fasano et al., 2023), con l'obiettivo di aggiornare i parametri di una distribuzione normale usata nuovamente come priori. I metodi verranno messi a confronto in termini di efficacia nell'individuare i giocatori più prestanti della stagione e in termini di carico computazionale.

Per tutte le analisi svolte è stato utilizzato il *software* statistico R (R Core Team, 2023, v: 4.3.1). I codici utilizzati sono disponibili in Appendice.

Capitolo 1

Il *dataset*

La tesi è focalizzata sulla modellazione di dati relativi a partite di pallacanestro del campionato NBA. Il primo obiettivo è creare una classifica della capacità offensiva dei giocatori aggiornata dinamicamente con il passare delle giornate di gioco. È necessario, dunque, fornirsi di dati che presentino un buon livello di dettaglio del contributo di ogni giocatore all'azione e dell'eventuale realizzazione del canestro. Le principali classifiche esistenti utilizzano statistiche descrittive che tengono conto di vari aspetti dei giocatori, come la percentuale di realizzazione, il numero di rimbalzi ottenuti o il numero di falli commessi. Esistono anche trasformazioni complesse che danno un'idea più generale dello stato di forma dei giocatori. Tuttavia, guardando solo a questi indici, non è facile determinare chi si sia realmente distinto per offensività, motivo che ha spinto la creazione di una classifica tramite modellazione statistica.

Nel presente capitolo verrà presentato il *dataset* utilizzato, definito *play-by-play*, poiché per ogni azione sono presenti tutti gli eventi e sotto-eventi che l'hanno caratterizzata, come un fallo, un tiro o un rimbalzo. Con questo tipo di informazione è possibile creare una valutazione del contributo di ogni giocatore alla realizzazione di punti durante le partite e, di conseguenza, un suo valore di pericolosità offensiva. È importante sottolineare che tra le variabili disponibili non è presente un indice di tale natura, il quale può tuttavia essere inferito mediante l'analisi di altre informazioni disponibili.

1.1 Fonte di dati

La ricerca di un *dataset* che rispetti i criteri appena descritti non è affatto banale. Al giorno d'oggi, l'utilizzo di dati e di tecniche statistiche in sport come la pallacanestro,

il calcio o il tennis è diventata una vera e propria disciplina. Di conseguenza, possedere informazioni sui campionati con un buon grado di dettaglio come quello del *play-by-play* è una notevole fonte di ricchezza e di conoscenza.

I dati a disposizione per il presente studio includono tutti gli incontri disputati dalla stagione 2004/05 fino alla stagione 2019/20, acquistabili al sito BigDataBall (2007). Tuttavia, si è deciso di concentrare l'attenzione solamente su una singola stagione, dato che da stagione a stagione i giocatori potrebbero aver avuto degli andamenti molto diversi, oltre ad aver potuto cambiare squadra, rendendo quindi difficile l'interpretazione dei risultati. Inoltre la scelta è giustificata anche dall'elevato numero di osservazioni presenti per ogni anno, motivo per cui si è ritenuta una singola stagione sufficiente.

Nonostante l'interesse fosse fornire una classifica quanto più aggiornata e recente, è stata esclusa dall'analisi la stagione 2019/20 concentrandosi solo su quella del 2018/19. Questo perché la stagione 2019/20 è stata fortemente influenzata dalla diffusione del COVID-19, costringendo i periodi di gara a continue interruzioni e assenze da parte dei giocatori, rendendo una valutazione temporale dei singoli *ranking* poco sensata.

1.2 Le variabili

I dati a disposizioni sono composti da 621527 osservazioni e relative 44 variabili. Ogni unità statistica si riferisce a un singolo evento avvenuto durante i 24 secondi che caratterizzano un'azione. Di conseguenza ogni azione è descritta da uno o più eventi, oltre che da un buon numero di informazioni aggiuntive, descritte in Tabella 1.1.

Un aspetto rilevante è la diversa natura delle variabili divise in:

- fattori con centinaia di modalità;
- variabili testuali;
- variabili in formato data o tempo;
- variabili numeriche.

Tutte queste caratteristiche permettono di affermare che si è in un caso di *big data* sia in volume, dato dall'alto numero di unità statistiche, sia in complessità, a causa della natura delle variabili concomitanti. Questi aspetti saranno particolarmente rilevanti in fase di modellazione, dato che uno dei principali ostacoli alle usuali tecniche statistiche risiede proprio nella dimensione del *dataset*.

Variabile	Descrizione	Tipo
game_id	Identificativo dell'incontro	Carattere
data_set	Tipo di campionato "Playoff" o "Regular Season"	Fattore (2 lv)
date	Data in cui si è svolto l'incontro	Data
a1, a2, a3, a4, a5	Variabili che indicano i 5 giocatori in campo fuori casa	Fattore (849 lv)
h1, h2, h3, h4, h5	Variabili che indicano i 5 giocatori in campo in casa	Fattore (849 lv)
period	Descrizione del periodo di gioco	Fattore (4 lv)
a_score, h_score:	Punteggio della squadra fuori casa e della squadra in casa	Numerica
r_time, elapsed	Tempo rimasto nel periodo e tempo giocato nel periodo	Tempo
play_length	Durata dell'evento	Tempo
play_id	Identificativo dell'evento	Carattere
team	Squadra a cui appartiene il giocatore che ha realizzato l'evento	Fattore (30 lv)
assist	Giocatore che ha svolto l'assist	Fattore (514 lv)
away, home	Giocatori che hanno partecipato alla palla a due	Fattore (318 lv)
block	Giocatore che ha eseguito un blocco	Fattore (475 lv)
entered, left	Giocatore che è entrato e che ha lasciato il campo	Fattore (509 lv)
num	Numero di tiro libero	Fattore (3 lv)
opponent	Giocatore che ha subito un fallo	Fattore (517 lv)
player	Giocatore che ha svolto l'evento	Fattore (849 lv)
points	Punti segnati nell'evento	Numerica
reason	Breve spiegazione dell'evento	Carattere
result	Risultato del tiro	Fattore (2 lv)
steal	Giocatore che ha rubato la palla	Fattore (507 lv)
event_type	Evento avvenuto	Fattore (15 lv)
type	sotto-evento avvenuto	Fattore (112 lv)
s_distance	Distanza del tiro in piedi	Numerica
o_x, o_y, c_x, c_y	Posizione dettagliata del tiro	Numerica
description	Descrizione dettagliata dell'evento	Carattere

Tabella 1.1: Variabili presenti nel *dataset*

1.3 Trattamento iniziale dei dati

Come accade spesso quando si ha a che fare con grandi masse di dati, le informazioni a disposizione potrebbero essere imprecise o affette da errori. Si è proceduto, di conseguenza, a una prima fase di pulizia e preparazione dei dati per le successive fasi di analisi.

In particolare, dato che l'attenzione si focalizza sui giocatori, ci si è accertato che i 10 giocatori in campo indicati dalle variabili `a1,...,a5, h1,...,h5` (Tabella 1.1) includessero il giocatore che ha svolto l'azione e che le variabili indicassero effettivamente 10 atleti diversi. Alcuni giocatori risultavano inoltre associati a squadre errate durante il primo evento di ogni partita ("jump_ball"). Sono state inoltre corrette alcune imprecisioni nelle variabili riferite al tempo e aggiustati degli errori testuali.

Si è deciso successivamente di eliminare tutte le righe a cui non venisse associata nessuna squadra e nessun atleta: questo perché riferite ad eventi non direttamente riconducibili ai giocatori, come un intervallo o l'inizio del periodo di gioco e, quindi, non utili a valutare l'efficacia offensiva dei singoli atleti.

Tutti i giocatori che presentassero un numero di eventi inferiore a 100 sono stati in seguito rimossi dal *dataset*, dato che, con un numero così basso di interventi in tutta la stagione, sarebbe risultato difficile avere delle valutazioni attendibili. Questa operazione ha ridotto notevolmente il numero di atleti totali presenti passando da 809 a 428, pur non riducendo drasticamente le dimensioni del *dataset*.

A questo punto è stata aggiunta un'informazione mancante ma di notevole importanza, ossia quella relativa al ruolo dei giocatori con la quale è possibile eseguire alcune analisi stratificate e valutare chi ha avuto la migliore *performance* condizionata al proprio ruolo. L'operazione è stata possibile tramite l'utilizzo di un *dataset* esterno e dell'utilizzo del pacchetto R `dplyr` (Wickham et al., 2023), il quale facilita notevolmente l'unione di diversi insiemi di dati.

Esistono essenzialmente 5 diversi ruoli nel *basket*, riassumibili in:

- **C**: centro, giocatore più alto e forte, di solito gioca in prossimità del canestro;
- **PG**: *play maker*, regista dell'attacco della squadra, ha la migliore visione di gioco;
- **SG**: guardia tiratrice, giocatore con un buon tiro in sospensione ma capace anche di penetrare la difesa;
- **SF**: ala piccola, giocatore versatile sia d'attacco che da difesa;

- **PF**: ala grande, giocatore fisico che deve prendere rimbalzi e difendere i giocatori più grandi.

Una volta conclusa questa prima fase di pulizia e correzione degli errori, si è proseguito con la creazione della matrice delle variabili esplicative utilizzate nella costruzione di modelli.

1.3.1 Evento e sotto-evento

Essendo l'obiettivo quello di valutare la pericolosità dei singoli giocatori è naturale immaginare un collegamento tra i vari tipi di evento e la successiva realizzazione di un canestro. Banalmente, i rimbalzi presi avranno un impatto positivo sull'aumentare della pericolosità dell'azione a differenza delle palle perse. Di conseguenza, per una prima fase di analisi, sono state utilizzate come variabili esplicative solamente quelle riferite a eventi e a sotto-eventi (Tabella 1.1), ricodificate in variabili *dummy*: per ogni livello delle variabili è stata realizzata una nuova variabile dicotomica uguale a 1 se presente quel determinato evento, 0 altrimenti. Il vantaggio in fase di modellazione e la maggior facilità interpretativa costituiscono i motivi essenziali della scelta.

Si è deciso di escludere dall'analisi alcune osservazioni con valori della variabile `event_type` poco interessanti per l'analisi, come l'inizio e la fine di un quarto, l'espulsione di un giocatore o valore mancante.

La variabile `type` ha richiesto invece un'analisi più approfondita, dato che fornisce un'informazione molto più dettagliata del tipo di evento dell'azione. Come prima cosa si è deciso di accorpare gli eventi considerati estremamente simili e di non considerare quelli troppo poco diffusi all'interno del *dataset* (più del 99.5% di 0). Inoltre sono state eliminate tutte le informazioni relative all'ordine di tiri liberi, considerate di poco interesse. In Tabella 1.2 è disponibile una descrizione più approfondita degli eventi e sotto-eventi.

Come operazione conclusiva, sono state aggiunte delle nuove righe con `event_type` uguale a "steal" per quelle osservazioni che presentavano la variabile `steal` diversa dal valore mancante, dato che questo tipo di evento era disponibile soltanto come variabile. In Tabella 1.3 viene riportato un esempio del procedimento. Di conseguenza, sono state eliminate tutte le righe dove la variabile originale `steal` presentava valore diverso da valore mancante (NA) per non creare informazione ridondante. Stessa procedura è stata adottata per la variabile `block`.

event_type	**type**
ejection: espulsione	ejection
foul: fallo commesso	13 diversi tipi di fallo tra cui: -offensive foul -technical foul
free throw: tiro libero	12 diverse informazioni sull'ordine del tiro libero tra cui: -free throw 1 of 2 -free throw 2 of 3
jump ball: palla a due	jump ball
miss: tiro sbagliato	38 diverse tipologie di tiro tra cui: -layup -jump shot
rebound: rimbalzo	3 diversi rimbalzi: -rebound offensive -rebound devensive -team rebound
sub: sostituzione	sub
timeout: intervallo	timeout
turnover: cambio possesso	26 tipi di cambio possesso tra cui: -bad pass -lost ball
violation: falli minori	6 diversi falli dovuti a violazioni minori: -kicked ball -lane
shot: tiro segnato	38 diverse tipologie di tiro tra cui: -Dunk -Hook shot

Tabella 1.2: Riassunto degli eventi e sotto-eventi

Un ragionamento analogo poteva essere fatto per la variabile `assist`, anch'essa non inclusa tra le modalità di `event_type`. Tuttavia, emerge che ad ogni *assist* corrisponde sempre un successivo canestro rendendo di fatto questa variabile una *leaker* della risposta. Per questo motivo si è deciso di escluderla dal *dataset* finale.

Al termine di queste operazioni, la matrice delle variabili esplicative è formata da

remaining_time	**elapsed**	**event_type**	**player**	**steal**
0:11:08	0:00:52	turnover	Ben Simmons	Gordon Hayward
0:11:08	0:00:52	steal	Gordon Hayward	NA

Tabella 1.3: Esempio di creazione di una nuova riga per l'evento "steal"

552179 osservazioni per 49 variabili indicatrici.

1.3.2 Creazione della variabile risposta

Chiarita la creazione delle variabili concomitanti è possibile ora discutere la scelta di una variabile risposta. Come anticipato, il *dataset* non presenta un indice di aggressività e di pericolosità dei giocatori in termini di contributo al canestro. In realtà non è così immediato immaginare una variabile che rispetti questi criteri e ancora più complicata è una sua misurazione diretta.

Seguendo l'idea proposta da Decroos et al. (2019) e successivamente ripresa da Dandolo (2019) e da Artuso (2020) si è deciso di considerare, all'interno di un'azione, un evento come pericoloso se nei successivi k eventi ha portato a un canestro. Di conseguenza, è stata creata una nuova variabile dicotomica con valore 1 se nei successivi k eventi consecutivi della squadra in possesso della palla è stato realizzato un canestro e 0 altrimenti.

La scelta del valore di k non è affatto banale: in sport come il calcio dove il numero di *goal* in un incontro è basso si utilizzano valori di k elevati, compresi tra i 10 e i 15, dato che gli eventi a disposizione tra i *goal* sono numerosi. In generale, valori alti di k considerano il contributo di eventi molto lontani dalla realizzazione di canestri (o *goal*) mentre valori piccoli di k si concentrano esclusivamente in eventi prossimi al canestro (o *goal*).

Il *basket* è uno sport estremamente dinamico; durante un incontro vengono realizzate anche decine e decine di canestri, il che rende le azioni estremamente rapide e povere di eventi. Esiste anche un limite regolamentare di 24 secondi per la conclusione dell'azione offensiva. Inoltre è bene sottolineare che, nonostante i dati utilizzati presentino un livello di dettaglio molto alto, non si hanno a disposizione tutti i singoli passaggi all'interno di un'azione, il che suggerisce di concentrarsi intorno a valori di k piccoli, compresi tra 3 e 5. Tuttavia, notando la proporzione di 1 della nuova variabile con i tre diversi valori di k proposti, non si notano delle evidenti differenze (Tabella 1.4). Questo significa che la maggior parte dei canestri è preceduta da massimo 3 eventi e che l'aggiunta del

k	**frazione di 1**
3	0.4102
4	0.4136
5	0.4149

TABELLA 1.4: Frazione di 1 nel campione per la variabile risposta con diversi valori di k

quarto e del quinto ritardo non fornisce ulteriore informazione. La scelta ricade dunque sull'utilizzo di $k = 3$.

Una volta creata la variabile risposta si avrà un vettore di dimensioni $n \times 1$ $y = (y_1, ..., y_n)$ formato da singole realizzazioni indipendenti di variabili casuali Bernoulli indipendenti di parametro ignoto, ossia

$$Y_i \sim Bern(\pi_i), \quad i = 1, ..., n,$$

dove π_i sarà modellato in funzione della variabili esplicative.

Ci si riconduce di conseguenza a un problema di classificazione binaria, dove l'obiettivo è quello di valutare la pericolosità dell'azione in base al tipo di evento o sotto-evento avvenuto. I metodi con cui si creerà il *ranking* dei giocatori a partire da questa variabile risposta verranno approfonditi nel Capitolo 2.

1.4 Indici statistici esistenti

Il mondo dell'NBA è caratterizzato dalla presenza di numerosi indici statistici che descrivono varie qualità dei giocatori. Alcune delle statistiche più comuni includono il numero di tiri, *assist*, rimbalzi, recuperi e stoppate. L'obiettivo del presente studio, come chiarito nel paragrafo precedente, è creare una classifica della pericolosità offensiva dei giocatori e del contributo alla realizzazione di un canestro. Si è deciso di riportare di seguito l'indice BPM (*Box Plus/Minus*, Myers 2020) essendo un indice estremamente completo e più sofisticato del classico PM (*Plus/Minus*), ossia la somma dei punti realizzati e subiti dalla squadra mentre il giocatore in oggetto è in campo. Il BPM normalizza il contributo del singolo giocatore su 100 possessi, in modo da avere una base comune a tutte le squadre. In questo modo si può effettuare un confronto diretto del contributo di tutti i giocatori. In sostanza il BPM fornisce i punti garantiti di un giocatore rispetto al livello medio della Lega, motivo per il quale è stato selezionato per eventuali confronti. Per i dettagli sul calcolo di questo indice si veda, ad esempio, Myers (2020).

Player	**Team**	**VORP**	**BPM**
James Harden	HOU	9.30	11.00
Giannis Antetokounmpo	MIL	7.40	10.40
Anthony Davis	NOP	5.30	9.40
Nikola Jokić	DEN	7.00	9.10
LeBron James	LAL	4.90	8.00
Paul George	OKC	6.60	7.20
Kyrie Irving	BOS	5.10	7.20
Kawhi Leonard	TOR	4.70	7.20
Stephen Curry	GSW	5.10	6.60
Nikola Vucević	ORL	5.5	6.60

Tabella 1.5: Valori dell'indice VORP e BPM per i primi 10 giocatori ordinati per BPM

In Tabella 1.5 sono riportati i primi 10 giocatori della stagione 2018/19 ordinati per BPM, disponibili in BasketballReference (2018). È inoltre riportato l'indice VORP (*Value Over Replacement Player*), anch'esso molto utilizzato nei *ranking* NBA, il quale permette di capire quale sia il contributo del giocatore in analisi paragonandolo al contributo di un giocatore di riferimento (*replacement player*). Il VORP tiene inoltre conto dei minuti e delle partite giocate. Si noti come, ad eccezione dei primi due protagonisti della classifica, i valori degli indici tra i giocatori non siano perfettamente ordinati, non rendendo facile eleggere un miglior giocatore in assoluto. Ogni indice ha un significato ben preciso, di conseguenza qualsiasi affermazione su un determinato *ranking* deve essere contestualizzata e presa con le giuste misure. Al primo posto della classifica si trova James Harden, giocatore che durante la stagione considerata ha realizzato 36.1 punti di media; non sorprende, di fatto, che la sua presenza in campo porti a un notevole incremento di punti per la squadra. Giannis Antetokounmpo, al secondo posto, ha vinto nell'annata 2018/19 il titolo di MVP delle Regular Season, il che giustifica la sua presenza nei gradini più alti del podio. Interessante notare come il terzo posto sia occupato da Anthony Davis. Nonostante sia stato uno dei migliori giocatori difensivi della stagione, l'indice suggerirebbe che sia stato fondamentale anche nell'incrementare lo *score* durante il campionato.

In generale, tutti i giocatori successivi possono vantare un'annata eccellente rendendo questa classifica attendibile per successivi confronti.

Capitolo 2

Approccio statico

Il presente capitolo si pone l'obiettivo di creare delle prime stime della pericolosità degli eventi tramite un approccio statico, ossia considerando, di giornata in giornata, i dati come indipendenti e di stimare un modello diverso solamente sui dati di ogni singola giornata. Si procederà poi alla creazione di un indice di pericolosità dei giocatori aggregando i risultati di ogni giornata di campionato. Lo *score* progressivo dei giocatori verrà quindi aggiustato per un particolare coefficiente e lisciato tramite una media mobile, in modo da fornire delle stime più attendibili e includere dell'informazione passata. In questo modo è possibile creare una classifica aggiornata di giornata in giornata e poter valutare l'andamento temporale degli atleti. Infine, i singoli *ranking* verranno stratificati per ruolo e per momento della stagione, poiché è risaputo che le prestazioni dei *top player* possono variare molto tra Regular Season e Playoff.

2.1 Modello *probit*

Dall'insieme delle variabili concomitanti descritto nel precedente capitolo è possibile creare la matrice del disegno X, di dimensione $n \times p$. Dato che tutte le esplicative considerate si presentano come variabili dicotomiche, la matrice X avrà una colonna in corrispondenza di ogni variabile più una colonna relativa all'intercetta e, di conseguenza, $p = 50$.

Indicato con y il vettore di dimensione $n \times 1$ con la variabile risposta, è possibile proporre un metodo che metta in relazione il vettore y e la matrice X.

Sia allora $y = (y_1, ..., y_n)$ realizzazione di $Y = (Y_1, ..., Y_n)$, con Y_i variabili casuali indipendenti $Bi(1, \pi_i)$ di parametro π_i ignoto compreso tra 0 e 1. Sia inoltre $\boldsymbol{x_i}$ il generico vettore di dimensione $p \times 1$ indicante la i-esima riga della matrice del disegno X e $\beta = (\beta_1, ..., \beta_p)$ un vettore di parametri p-dimensionale.

È possibile mettere in relazione le probabilità di successo π_i con le variabili esplicative $\boldsymbol{x_i}$ e il vettore β secondo la relazione

$$g(\pi_i) = \eta_i = \boldsymbol{x_i}^T \beta = \sum_{r=1}^{p} \beta_r x_{ir}, \quad i = 1, ..., n.$$

Poiché π_i assume valori nell'intervallo (0,1), come funzione di legame $g(\cdot)$ si sceglie, in genere, una funzione $g : [0, 1] \to \mathbb{R}$ monotona crescente. Per il presente studio si è scelto di utilizzare un modello *probit*, che corrisponde a scegliere come funzione legame l'inversa della funzione di ripartizione della normale standard:

$$g(\pi_i) = \Phi^{-1}(\pi_i) = \boldsymbol{x_i}^T \beta, \tag{2.1}$$

dove $\Phi(\cdot)$ rappresenta la funzione di ripartizione della normale standard. Il motivo di tale scelta verrà discusso nel Capitolo 3. Un'altra scelta comune in casi di classificazione binaria è il modello *logit*, il quale assume come funzione di legame $g(\pi_i) = \log\{\pi_i/(1 - \pi_i)\}$. Il modello *logit* e il modello *probit* portano tipicamente a risultati simili se le vere probabilità π_i non sono troppo vicine alla frontiera dello spazio parametrico. Per l'utilizzo di altre funzioni di legame si veda, ad esempio, Salvan et al. (2020).

Isolando π_i in (2.1) si ottiene

$$\pi_i = \Phi(\boldsymbol{x}_i^T \beta) = \Phi(\eta_i).$$

Essendo le singole osservazioni indipendenti, la funzione di verosimiglianza sarà dunque

$$L(\beta; X, y) = L(\beta) \propto \prod_{i=1}^{n} \Phi(\eta_i)^{y_i} \{1 - \Phi(\eta_i)\}^{1-y_i}. \tag{2.2}$$

Tramite la massimizzazione della funzione (2.2) è possibile ricavare la stima di massima verosimiglianza $\hat{\beta}$ del vettore di parametri β

$$\hat{\beta} = \underset{\beta}{\text{argmax}} \; L(\beta; X, y). \tag{2.3}$$

La soluzione della (2.3) non è ricavabile per via analitica, rendendo necessario l'utilizzo di metodi iterativi, nella fattispecie una variante del metodo di Newton-Raphson chiamato Minimi Quadrati Pesati Iterati (si veda ad esempio Salvan et al., 2020, Paragrafo 2.3.6).

Come primo passo, viene dunque adattato il modello appena descritto su tutti i dati a disposizione, con lo scopo di avere delle prime valutazioni grezze delle scelte fatte riguardo alle variabili esplicative, alla scelta della variabile risposta fino al modello *probit*. È importante ricordare che l'obiettivo della tesi è di creare una classifica dei giocatori

dinamica aggiornata con il susseguirsi delle partite. Di conseguenza, adattare un modello sull'intero *dataset* senza considerare alcun ordinamento temporale non risponde alla domanda di interesse.

Una volta adattato il modello ne sono state valutate le *performance* in termini di tasso di corretta classificazione, sensibilità e specificità. Il tasso di corretta classificazione rappresenta il numero di osservazioni correttamente classificate sul totale delle osservazioni. La sensibilità indica il numero di osservazioni correttamente classificate come 1 sul totale degli 1 presenti nel campione mentre per la specificità vale lo stesso principio per gli 0. Dato che, come emerge in Tabella 1.4, la percentuale di 0 e 1 presente non è perfettamente bilanciata sono state valutati diversi valori della soglia, a cominciare dalla proporzione di 1 nel campione. In Tabella 2.1 è possibile osservare le prestazioni del modello al variare della soglia. Il valore più appropriato in termini di corretta classificazione sembrerebbe aggirarsi intorno a 0.5, tuttavia con questo valore soglia è chiaro come il modello fatichi ad individuare correttamente gli 1 notando al basso valore della sensibilità, motivo che ha spinto a scegliere un valore della soglia che permettesse ai valori di sensibilità e specificità di essere più simili tra loro, pari a 0.4.

soglia	**t_corretta**	**sensibilità**	**specificità**
0.3	0.6111	0.9367	0.3847
0.35	0.6115	0.9359	0.3859
0.4	0.6864	0.6602	0.7046
0.45	0.6991	0.5448	0.8065
0.5	0.7035	0.4691	0.8665
0.55	0.7029	0.4344	0.8896

Tabella 2.1: Prestazioni del modello *probit* al variare della soglia

Con il livello di soglia fissato, si ottiene una tabella di corretta classificazione visibile in Tabella 2.2.

	Valori osservati	
Valori previsti	0	1
0	282208	120237
1	43480	106254

Tabella 2.2: Matrice di corretta classificazione ottenuta con valore della soglia pari a 0.4

Il modello non presenta dei risultati particolarmente entusiasmanti, tuttavia la percentuale di falsi positivi e la percentuale di falsi negativi sembrerebbe essere abbastanza

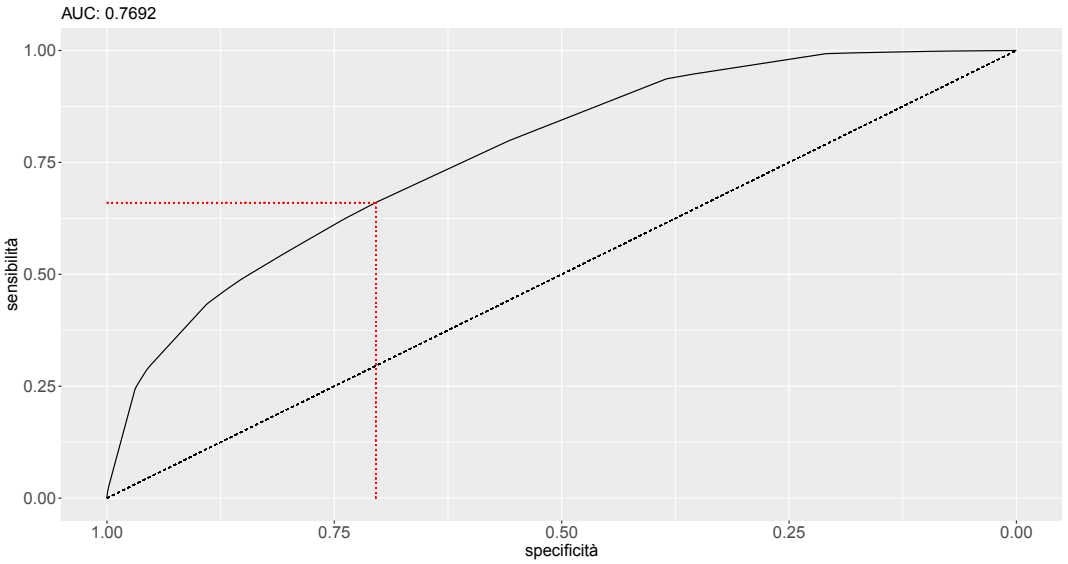

FIGURA 2.1: Curva ROC per il modello *probit* adattato

simile, suggerendo che il metodo non ha particolari sbilanciamenti verso una delle due modalità della risposta. La curva ROC in Figura 2.1 conferma le discrete prestazioni del modello al variare dei valori soglia.

In conclusione, viene ritenuto accettabile questo tipo di approccio come punto di partenza per i successivi *step* della creazione del *ranking*, approfondito nei paragrafi successivi.

2.1.1 Modello *stepwise*

Prima di procedere con l'adattamento del modello di classificazione di giornata in giornata, si è deciso di eseguire una procedura di *stepwise backward* sul modello che include tutte le osservazioni, con lo scopo di eliminare quelle variabili non utili ai fini dell'analisi. La scelta è stata dettata dalla presenza di alcuni eventi rari e di altri estremamente simili tra loro. Di conseguenza si è optato per una procedura di selezione automatica che ne eliminasse alcune.

In particolare, il procedimento *stepwise* con direzione *backward* valuta la variazione dell'indice AIC (*Akaike's information criterion*), presentato da Akaike (1973), al diminuire graduale delle variabili esplicative all'interno del *dataset*. È noto che

$$AIC = -2\{l(\hat{\beta}; X, y) - p\}, \tag{2.4}$$

dove $l(\hat{\beta}; X, y)$ è il logaritmo della funzione di verosimiglianza, detta log-verosimiglianza, presentata in (2.2) valutata in $\hat{\beta}$, mentre p indica il numero di parametri presenti. La funzione di verosimiglianza assume valore massimo in $\hat{\beta}$ ed è noto che questo valore

	Valori osservati	
Valori previsti	0	1
0	229466	76967
1	96222	149524

soglia	0.4
t_corretta	0.6863
sensibilità	0.6601
specificità	0.7046

Tabella 2.3: Matrice di corretta classificazione e prestazioni del modello *stepwise*

aumenta all'aumentare del numero di parametri. L'obiettivo è quello di minimizzare l'indice, dunque per non permettere al valore dell'AIC di diminuire costantemente all'aumentare del numero di parametri, viene introdotta un penalità pari a p, in modo tale da trovare un compromesso tra l'adattamento del modello e la sua complessità. La formula è ottenibile tramite la minimizzazione della divergenza di Kullback-Leibler tra la vera distribuzione dei dati e il modello stimato (si veda ad esempio Azzalini & Scarpa, 2012, Paragrafo 3.5.3). Il termine moltiplicativo -2 è stato inserito per allinearsi a notazioni legate alla verosimiglianza, in particolare il *test* del log-rapporto di verosimiglianza.

Procedendo con la stima, al primo *step* dell'algoritmo verrà esclusa una variabile alla volta e stimato ogni modello ridotto privo della determinata variabile candidata a essere esclusa. Il modello che porterà al maggior decremento dell'indice AIC rispetto al modello completo verrà selezionato come modello successivo. Si procederà così fino a quando l'omissione di qualsiasi variabile non porterà a un miglioramento in termini di AIC.

A fine procedura sono state eliminate 7 variabili, riducendo il totale di esplicative da 49 a 42: `rebound_offensive`, `turnover`, `traveling`, `off_foul`, `kicked_ball`, `offensive_charge_foul`, `personal_take_foul`.

Stimando nuovamente il modello descritto nel Paragrafo 2.1 è interessante notare dalla Tabella 2.3 come le conclusioni siano praticamente identiche a quelle del modello completo, giustificando la semplificazione.

Le nuove variabili selezionate verranno utilizzate nei modelli stimati di giornata in giornata. Tuttavia, è possibile che in alcune giornate le variabili escluse dall'analisi siano invece rilevanti. Una procedura più corretta prevedrebbe di stimare, di giornata in giornata, il modello più adatto tramite una procedura *stepwise*. Purtroppo questa procedura non è praticabile a causa del notevole costo computazionale del metodo e dell'elevato numero di modelli da adattare ad ogni *step*, motivo per il quale si è deciso di utilizzare le variabili selezionate sulla totalità dei dati come variabili da utilizzare di giornata in giornata.

2.1.2 Inserimento effetti fissi

A questo punto, si dispone di un considerevole numero di variabili esplicative. Tuttavia, il *dataset* contiene anche informazioni aggiuntive rilevanti per valutare le prestazioni individuali dei giocatori. In particolare fin'ora non è stata inclusa la variabile relativa al ruolo, inserita manualmente da un *dataset* esterno. Di conseguenza, per non perdere informazione relativa alla posizione e per fornire delle valutazioni realistiche di quanto un giocatore possa incidere nel gioco dato il suo ruolo, viene inserito un effetti fisso per questa variabile.

È inoltre logico pensare di inserire un effetto fisso per ogni singolo giocatore, in modo da non trascurare le sue caratteristiche individuali e includere le differenze intrinseche tra i giocatori stessi. Tuttavia, l'inserimento di un numero così alto di parametri aggiuntivi porta a un evidente appesantimento delle procedure di stima. Un'alternativa è quella di includere degli effetti casuali per i singoli giocatori. Tuttavia gli effetti casuali si basano sull'indipendenza tra gli errori a essi associati e le restanti variabili esplicative inserite nel modello, assunzione facilmente confutabile. Si è quindi deciso, motivato anche dall'elevato numero di osservazioni presenti nel *dataset*, di inserire un effetto fisso per ogni giocatore.

L'inserimento delle due informazioni appena descritte porta all'aggiunta di un numero notevole di parametri, dato che sono presenti 5 ruoli e 428 giocatori. Il numero totale di parametri da stimare dipende però anche da vincoli lineari tra le colonne della matrice del disegno. Un'esemplificazione del problema è disponibile in Tabella 2.4, nella quale sono presenti per dodici finti eventi solamente tre ruoli e sei giocatori e la relativa matrice del disegno. In questo caso il ruolo "A" e il giocatore "pl_1" sono state prese come modalità di riferimento, seguendo l'ordine alfabetico.

Dato che ogni giocatore occupa un solo ruolo è chiaro che lo schema così descritto presenta delle colonne linearmente dipendenti: ad esempio la colonna del "pl_3" sommata alla colonna del "pl_4" fornisce esattamente la colonna del "r_B". Di conseguenza, per ogni singolo ruolo, ad eccezione di quello appartenente al giocatore di riferimento (in questo caso il ruolo "A"), verrà escluso un giocatore per permettere la stima di tutti i parametri. Il predittore lineare del giocatore escluso è comunque ricavabile ponendo tutte le indicatrici uguali a zero tranne quella relativa al suo ruolo.

Nel *dataset* utilizzato, i giocatori per cui non è stimato un parametro diretto sono: "Aaron Gordon" (riferimento), "Zaza Pachulia", "Yogi Ferrell", "Zach LaVine", "Yuta Watanabe", dove gli ultimi quattro atleti si trovano in ultima posizione, in ordine alfabetico, all'interno del proprio ruolo.

ruolo	player
A	pl_1
A	pl_1
A	pl_2
A	pl_2
B	pl_3
B	pl_3
B	pl_4
B	pl_4
C	pl_5
C	pl_5
C	pl_6
C	pl_6

Intercetta	r_B	r_C	pl_2	pl_3	pl_4	pl_5	pl_6
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	0	0	1	0	0
1	1	0	0	0	1	0	0
1	0	1	0	0	0	1	0
1	0	1	0	0	0	1	0
1	0	1	0	0	0	0	1
1	0	1	0	0	0	0	1

Tabella 2.4: Esemplificazione dello schema degli effetti fissi inseriti nel modello

Il numero totale di parametri da stimare viene dunque modificato, passando da 50 a 470, con 423 parametri relativi ai giocatori e 4 relativi alle posizioni.

Oltre a giustificare questa operazione da un punto di vista logico è possibile capire se ci sia dell'evidenza statistica a favore dell'inserimento degli effetti fissi. Una possibilità è quella di eseguire un *test* del log-rapporto di verosimiglianza tra il modello completo (con effetti fissi) e il modello ridotto (senza effetti fissi). Il risultato è visibile in Tabella 2.5. La differenza di devianze residue tra i due modelli confrontata con un chi-quadro con 427 gradi di libertà suggerisce evidenza verso il modello completo, dato il valore del *p-value* prossimo allo zero. L'inserimento di un termine fisso per ruolo e giocatore sembra dunque appropriato.

	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
1	552136	601951.01			
2	551709	600462.74	427	1488.27	2.2e-16

Tabella 2.5: Log rapporto di verosimiglianza tra il modello completo e il modello ridotto

In Tabella 2.6 sono riportati tutti i coefficienti stimati dal modello *probit* con i soli eventi significativi individuati con la procedura *stepwise* ad eccezione degli effetti fissi sui giocatori e sui ruoli.

Si nota che, commettere un fallo porta a una diminuzione della probabilità di essere in un'azione pericolosa, a differenza di, come ci si aspetterebbe, dei tiri liberi e degli `Shot`, che portano invece un incremento al netto delle restanti variabili. È importante

	Estimate	**Std. Error**	**z value**	**Pr(>|z|)**
foul	-0.28092	0.04051	-6.93464	4.07e-12
free_throw	0.72113	0.00848	85.04137	0.00000
jump_ball	1.46126	0.03748	38.99242	0.00000
rebound	1.66670	0.02541	65.60060	0.00000
Alley_Oop_Dunk	0.98525	0.03937	25.02513	0.00000
Alley_Oop_Layup	0.17605	0.04017	4.38264	1.17e-05
bounds_turnover	-0.10742	0.05432	-1.97754	4.79e-02
Driving_Finger_Roll_Layup	0.24099	0.02073	11.62755	2.98e-31
Driving_Hook_Shot	-0.26577	0.03309	-8.03172	9.61e-16
Driving_Reverse_Layup	0.05176	0.02772	1.86700	6.19e-02
Dunk	0.86210	0.03346	25.76787	0.00000
Fadeaway_Jumper	-0.42400	0.02115	-20.04750	2.12e-89
Finger_Roll_Layup	0.35687	0.05957	5.99122	2.08e-09
Floating_Jump_Shot	-0.29962	0.01877	-15.96500	2.24e-57
Hook_Shot	-0.30070	0.02204	-13.64027	2.31e-42
Jump_Bank_Shot	-0.14918	0.04127	-3.61500	3.00e-04
Jump_Shot	-0.54326	0.00740	-73.41102	0.00000
l_b_foul	0.15775	0.05591	2.82135	4.78e-03
Layup	-0.25636	0.01322	-19.38804	9.73e-84
lost_ball	0.35406	0.09725	3.64078	2.71e-04
offensive_foul_turnover	-0.12326	0.04447	-2.77149	5.58e-03
p_foul	0.12263	0.03785	3.24004	1.19e-03
Pullup_Jump_Shot	-0.41159	0.01001	-41.13055	0.00000
Putback_Dunk	1.03871	0.07641	13.59375	4.36e-42
Putback_Layup	0.29004	0.02634	11.01001	3.42e-28
rebound_defensive	-0.26690	0.00907	-29.42412	0.00000
Reverse_Layup	0.09620	0.03151	3.05271	2.26e-03
Running_Dunk	1.51316	0.05509	27.46910	0.00000
Running_Finger_Roll_Layup	0.55412	0.04798	11.54818	7.54e-31
Running_Jump_Shot	-0.40471	0.03404	-11.88790	1.36e-32
Running_Layup	0.23659	0.01990	11.88831	1.36e-32
Running_Reverse_Layup	0.52442	0.07556	6.94075	3.90e-12
s_foul	-0.20777	0.03958	-5.24947	1.52e-07
Step_Back_Jump_Shot	-0.40443	0.01515	-26.68711	0.00000
Turnaround_Fadeaway	-0.36913	0.02504	-14.74010	3.56e-49
Turnaround_Hook_Shot	-0.21550	0.02498	-8.62791	6.25e-18
Turnaround_Jump_Shot	-0.45848	0.02265	-20.24316	4.08e-91
defensive_goaltending	1.31888	0.05926	22.25667	0.00000
steal	1.58136	0.02589	61.07447	0.00000
block	-1.11624	0.01440	-77.51455	0.00000
sub	0.84736	0.02489	34.04897	0.00000
Shot	2.04155	0.02478	82.38966	0.00000

Tabella 2.6: Coefficienti stimati dal modello *probit* su tutte le osservazioni

sottolineare che conclusioni marginali sulle singole variabili devono essere prese con estrema cautela dato che, per alcuni gruppi di eventi, non è possibile considerare l'effetto di una variabile fermo restando le rimanenti. Ad esempio, il `p_foul` fa parte della categoria dei `foul`, dunque le due variabili sono fortemente legate, come i diversi tipi di tiro e la variabile `Shot`.

2.2 Modello *probit* con correzione di Firth

Il modello corrente mette a disposizione un buono strumento per valutare la pericolosità di un'azione a partite dall'evento e dai giocatori. Tuttavia, come ampiamente discusso in precedenza, l'obiettivo è quello di aggiornare la classifica degli atleti man mano che le giornate di gioco avanzano. Una possibilità è quella di stimare da capo il modello *probit* ad ogni nuova giornata di gioco utilizzando tutti i dati disponibili fino a quel momento. Così facendo sorgono però due problemi: il primo di natura computazionale, dato che il gran numero di parametri e l'aumentare delle osservazioni richiedono parecchio tempo; mentre il secondo di natura logica: in questo modo infatti si andrebbero a pesare allo stesso modo eventi passati con eventi recenti, creando di fatto una classifica non aggiornata e poco credibile. L'idea viene dunque scartata.

Una strada alternativa suggerisce invece di stimare un nuovo modello *probit* per ogni giornata di gioco utilizzando solamente i dati di quella determinata giornata. Di conseguenza, si considerano le giornate indipendenti, dato che i modelli non interagiscono tra di loro e vengono stimati da capo ogni volta che arrivano nuovi dati. Le previsioni della variabile risposta saranno fatte di giornata in giornata sui soli dati su cui è stimato il generico modello. In questo modo sorge tuttavia un ulteriore problema, legato al basso numero di osservazioni disponibili per ogni giornata. Quando il numero di parametri cresce avvicinandosi al numero di osservazioni è noto che le proprietà dello stimatore di massima verosimiglianza vengono meno (Zhao et al., 2020). In particolare, utilizzando una sola giornata di gioco alla volta, gli *standard error* dei coefficienti stimati risultano molto alti fornendo dei risultati non affidabili. Inoltre le stime della probabilità di pericolosità dell'azione sono molto vicine alla frontiera dello spazio parametrico. Queste diagnosi suggeriscono che si è in un caso di perfetta separazione (si veda ad esempio Agresti, 2015, Paragrafo 5.4.2), ossia l'iper-piano di dimensione p appartenente allo spazio delle variabili esplicative riesce a dividere perfettamente le osservazioni con risposta uguale a 1 da quelle con risposta uguale a 0. È necessario dunque pensare a un metodo che vada oltre i limiti presentati.

Si propone qui di utilizzare la correzione per la distorsione introdotta da Firth (1993) che consiste nell'effettuare una modifica della funzione punteggio con l'obiettivo di ottenere uno stimatore con distorsione di ordine ridotto rispetto a quella dello stimatore di massima verosimiglianza.

Sia $U(\beta)$ la funzione punteggio relativa alla verosimiglianza (2.2) definita come

$$U(\beta) = \frac{\partial}{\partial\beta} l(\beta).$$

È noto che è possibile ricavare la stima di massima verosimiglianza $\hat{\beta}$ risolvendo la cosiddetta equazione di verosimiglianza $U(\beta) = 0$. Firth (1993) propone di inserire una piccola distorsione nella funzione punteggio $U(\beta)$. In particolare, sia $b(\beta)$ il termine dominante della distorsione dello stimatore di massima verosimiglianza e sia

$$i(\beta) = -\mathbb{E}_\beta \left[\frac{\partial}{\partial\beta^T} U(\beta) \right]$$

la matrice di informazione di Fisher. È possibile definire la nuova funzione punteggio come

$$U^*(\beta) = U(\beta) - i(\beta)b(\beta) \tag{2.5}$$

e risolvendo per β la nuova equazione di stima

$$U^*(\beta) = 0_p \tag{2.6}$$

si otterrà uno stimatore β^* la cui distorsione sarà minore rispetto a quella di $\hat{\beta}$. Supponendo di trovarsi nel caso di una famiglia esponenziale con *link* canonico, dunque *link logit* in questo caso, grazie ai contributi di Firth (1993) la soluzione dell'equazione appena presentata corrisponde a massimizzare la funzione di verosimiglianza penalizzata

$$L^*(\beta) = L(\beta)|i(\beta)|^{\frac{1}{2}}. \tag{2.7}$$

È interessante notare che $|i(\beta)|^{\frac{1}{2}}$ rappresenta la distribuzione a priori di Jeffreys (Jeffreys, 1946), il che permette di dare un'interpretazione bayesiana al metodo. La (2.7) è infatti massimizzata in β^* che è la moda a posteriori con priori di Jeffreys. Ricordando che, nel caso ancora di un modello di regressione binaria con *link* canonico, la matrice di informazione osservata può essere scritta come

$$i(\beta) = X^T W X, \qquad \text{con} \quad W = \text{diag}(w_1, ..., w_n), \tag{2.8}$$

dove il generico elemento w_i è uguale a $\pi_i(1-\pi_i)$. È possibile dimostrare che il determinante di $i(\beta)$ è massimizzato quando è massimo ogni w_i, ovvero quando $\pi_i = 0.5$ e di conseguenza quando $\beta = 0_p$. Pesare la verosimiglianza per la priori di Jeffreys corrisponde quindi a inserire un effetto di *shrinkage* per i β, comprimendoli verso lo 0.

Discorso analogo vale per il modello *probit* per il quale, tuttavia, risolvere l'equazione di stima (2.6) non corrisponde a massimizzare la verosimiglianza penalizzata in (2.7). La quantità per la quale viene penalizzata la verosimiglianza non assume una forma nota, ma è ancora non informativa e mantiene la proprietà di *shrinkage* dei parametri verso lo 0 (Kosmidis & Firth, 2020). In questo modo si crea una soluzione al problema della perfetta separazione indicato precedentemente.

Per l'implementazione del metodo è stato utilizzato il pacchetto R `brglm2` (Kosmidis, 2020), con il quale è possibile calcolare la soluzione dell'equazione di stima appena presentata.

2.2.1 Costruzione del *ranking*

Si hanno ora a disposizione, per ogni giornata di gioco in cui si è svolta almeno una partita, le previsioni della pericolosità delle azioni date dal modello *probit* con correzione di Firth. È possibile dunque creare un indice di pericolosità per i singoli giocatori. Una possibilità è quella di aggregare gli esiti previsti della variabile risposta, 0 o 1, secondo qualche regola, come ad esempio una somma o una media. Tuttavia, in questo modo si è fortemente vincolati dal valore della soglia scelto, un aspetto non banale da valutare di giornata in giornata. Di conseguenza, si è deciso di utilizzare le stime di probabilità di pericolosità dell'azione fornite dal modello e non la loro versione troncata a 0 o 1, mantenendo così più informazione. Per ogni giocatore verrà calcolata la media delle pericolosità fornite dal modello negli eventi in cui ha partecipato, per ogni giornata.

Formalmente, sia $\hat{\pi}_{j,t} = (\hat{\pi}_{1,j,t}, .., \hat{\pi}_{n_{jt},j,t})$ il vettore di probabilità contenente tutte le stime fornite dal modello *probit* per il giocatore j-esimo stimato nella giornata t-esima, con $j = 1, .., J$ e $t = 1, ..., T$. Una prima stima della pericolosità del giocatore j-esimo nella t-esima giornata può essere ricavata come

$$\hat{R}_{j,t} = \frac{1}{n_{jt}} \sum_{i=i}^{n_{jt}} \hat{\pi}_{i,j,t}, \tag{2.9}$$

dove n_{jt} indica il numero di eventi svolti dal giocatore j nella t-esima partita. Logicamente, i giocatori che non hanno effettuato tocchi di palla nella giornata t-esima non avranno un valore dell'indice associato a quella data. Si è inoltre deciso di non fornire

la stima per quei giocatori con un numero di eventi per giornata inferiore a 5 per non riportare conclusioni supportate da pochi dati. Questa prima stima grezza dell'offensività dei giocatori verrà perfezionata nei successivi paragrafi.

2.2.2 *Ranking* corretto

È importante sottolineare che le stime fornite dal modello tengono conto soltanto del giocatore, del suo ruolo e dell'evento in cui è coinvolto. Non sono dunque stati presi in considerazione né il numero di eventi di cui il giocatore è protagonista né il numero di partite che ha disputato nella stagione. Questi aspetti sono invece da tenere in considerazione per la creazione del *ranking*: a parità di punteggio è logico premiare quei giocatori con un alto numero di eventi, dato che ricoprono un ruolo più di spicco all'interno della squadra. Va anche considerato, come anticipato prima, il numero di partite disputate dato che i giocatori con più presenze hanno sicuramente più azioni.

Valutando questi aspetti, si è deciso di pesare il *ranking* creato, visibile in (2.9), per il numero di eventi di cui il j-esimo giocatore è stato protagonista fino a quel momento sul numero di minuti in cui è stato in campo. I minuti di gioco forniscono informazione aggiuntiva sulla presenza in campo dei giocatori rispetto alle sole partite, poiché è possibile che i minuti giocati siano molto variabili da partita a partita.

In pratica, il punteggio corretto avrà una forma del tipo

$$\widetilde{R}_{j,t} = \hat{R}_{j,t} \sum_{i=1}^{t} \frac{n_{ji}}{min_{ji}},$$

dove min_{ji} rappresenta il numero di minuti giocati nella i-esima partita dal giocatore j. In questo modo si pesa il coefficiente iniziale per il numero di azioni al minuto ottenuto fino a quel giorno. Dalla formulazione appena descritta, avere partecipato a un alto numero di azioni rappresenta un fattore di forza.

Tuttavia non tutti gli eventi portano a un aumento della probabilità di ottenere un canestro, come è chiaro dalla Tabella 2.6. In realtà, è molto raro che giocatori che rappresentano un ruolo di spicco per la squadra siano protagonisti solo di eventi sfavorevoli, di conseguenza si può considerare la quantità come una buona modifica del *ranking* iniziale. In Figura 2.2 sono riportati i 4 coefficienti stimati per 4 diversi giocatori del campionato. Si è voluto confrontare due giocatori estremamente forti con altri due meno conosciuti, in modo tale da capire se il coefficiente fosse in grado di cogliere queste differenze. È evidente che Giannis Antetokounmpo e Joel Embiid presentano un valore del coefficiente molto più alto rispetto agli altri due giocatori, oscillando intorno a valori

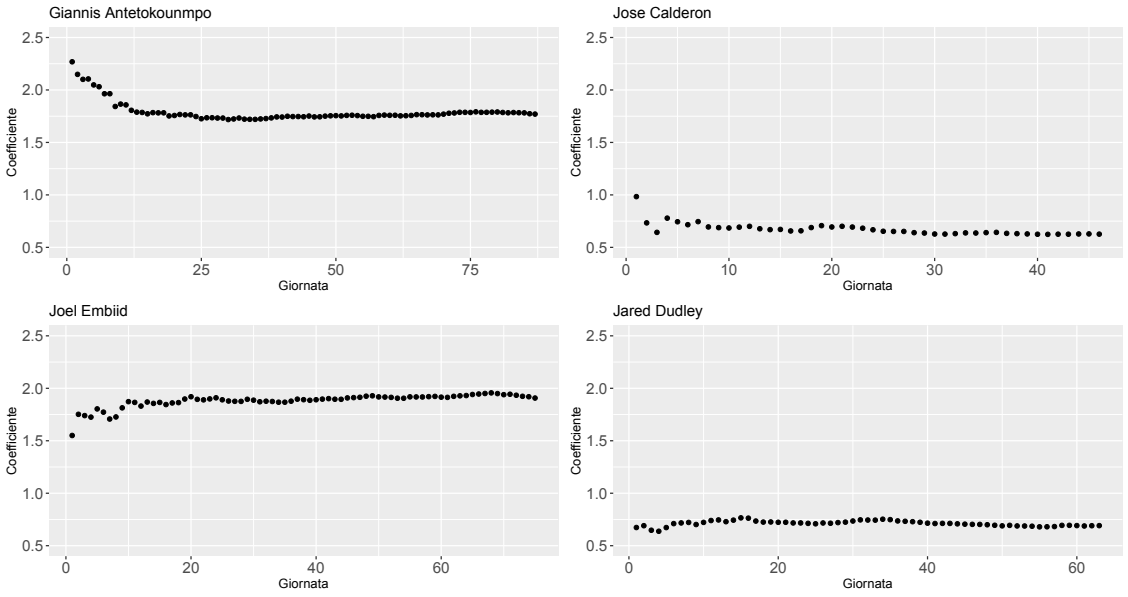

FIGURA 2.2: Andamento dei coefficienti di aggiustamento per 4 diversi giocatori

uguali a 2 a differenza degli altri due atleti con valore intorno allo 0.75. Il metodo sembrerebbe dunque essere adatto a cogliere differenze tra le abilità dei giocatori.

Si noti come, dopo una prima fase incerta caratterizzata da qualche oscillazione, il *trend* dell'evoluzione dei coefficienti sembrerebbe stabilizzarsi. Questo è dovuto alla sua natura, generato dal rapporto di due somme cumulate, le quali tendono a stabilizzarsi con il susseguirsi di un numero elevato di partite.

2.2.3 Lisciamento a media mobile

A questo punto si ha a disposizione una sequenza di punteggi in corrispondenza delle partite in cui ogni giocatore ha partecipato corretta per il coefficiente presentato nel Paragrafo 2.2.2. Nonostante il coefficiente d'aggiustamento rappresenti il numero di eventi al minuto dall'inizio della stagione e includa dunque nella sua definizione della dipendenza temporale, si è deciso di approfondire quest'ultimo aspetto. Specialmente a fine campionato, come chiaro in Figura 2.2, il coefficiente tende a stabilizzarsi, di conseguenza l'inclusione di dipendenza temporale non è più così forte.

Si è dunque deciso di applicare una media mobile pesata non centrata alla serie di punteggi ottenuta. In questo modo si include negli *score* una forte dipendenza dallo stato di forma delle giornate precedenti a quella presa in considerazione. La scelta del numero di giornate da considerare e il peso da attribuire a ciascuna di esse non è un problema banale. Dando troppa importanza a punteggi passati si rischierebbe di creare delle serie troppo lisce, non in grado di cogliere giornate negative. Al contrario, pesi

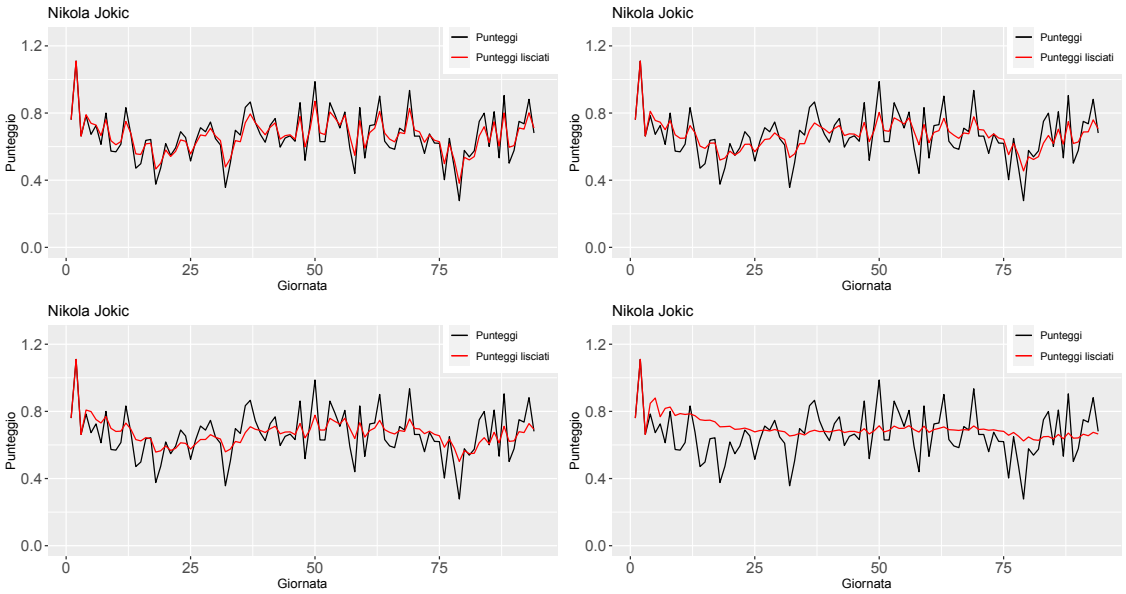

FIGURA 2.3: Serie dei punteggi e rispettiva serie lisciata dalla media mobile per 4 diversi vettori di pesi: (0.1, 0.1, 0.2, 0.6), (0.1, 0.2, 0.3, 0.4), (0.2, 0.2, 0.3, 0.3), (0.4, 0.3, 0.2, 0.1) partendo da in alto a sinistra e spostandosi per riga

troppo elevati per la partita in considerazione rischierebbero di creare una curva molto frastagliata, caratterizzata da continui bruschi cambi di direzione. Si è deciso dunque di valutare, per lo più graficamente, diversi possibili pesi da applicare alla media mobile e alcuni diversi ritardi. Si è scelto di considerare le 3 partite precedenti quella di interesse. Per la scelta dei pesi è possibile osservare la Figura 2.3, riferita al giocatore Nikola Jokic. È chiaro che assegnare più peso alla partita corrente favorisce bruschi cambi di direzione alla curva risultante la quale non presenta un netto discostamento dalla serie originale, come si può vedere nel grafico in alto a sinistra, per il quale sono stati utilizzati i pesi (0.1, 0.1, 0.2, 0.6). Discorso opposto vale per il grafico in basso a destra, generato considerando il vettore (0.4, 0.3, 0.2, 0.1). In quest'ultima, la maggior parte del peso è assegnata alle 3 partite precedenti quella di interesse, di conseguenza la serie finale risulta molto liscia. Si noti come con questi pesi la curva in rosso non riesca a cogliere le brutte prestazioni del giocatore Nikola Jokic intorno alla settantesima partita di campionato, oltre a fornire punteggi troppo ottimistici per le prime venti partite, dovuti probabilmente al picco visibile in corrispondenza del secondo giorno. Si è valutata dunque questa combinazione di pesi poco adeguata per lo scopo della tesi. La scelta è ricaduta nel vettore (0.1, 0.2, 0.3, 0.4) considerato un buon compromesso tra le due situazioni estreme appena presentate. Come si nota infatti nel grafico in alto a destra, la curva rossa riesce a seguire con efficacia il *trend* della curva nera creandone una versione più liscia.

Il nuovo punteggio può essere formalmente riassunto come

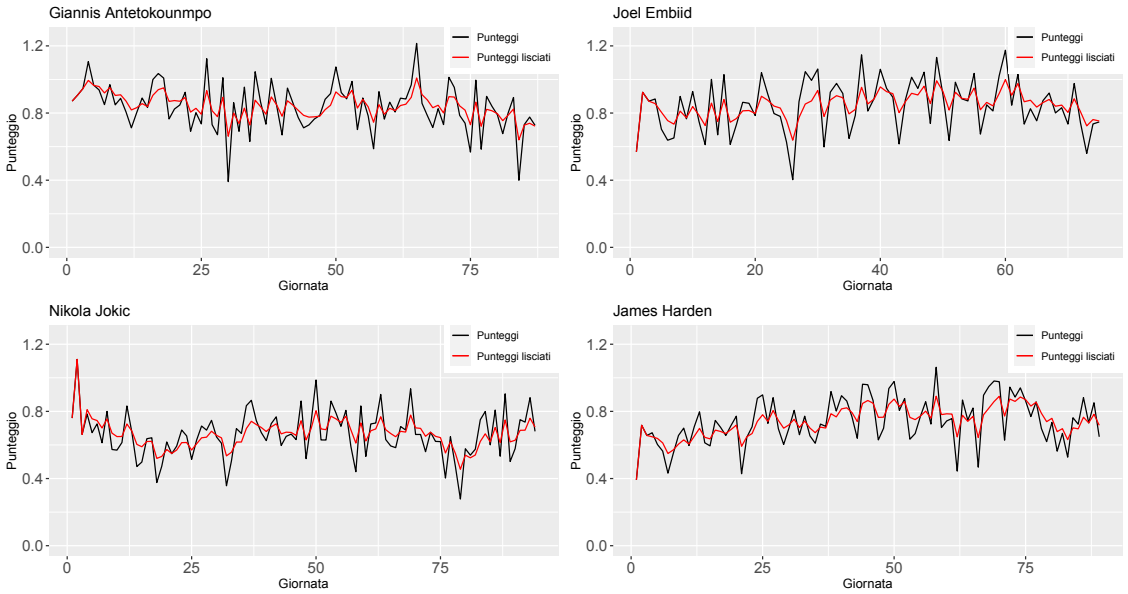

FIGURA 2.4: Andamento del *ranking* di 4 giocatori durante la stagione

$$R^*_{j,t} = \begin{cases} \widetilde{R}_{j,t} & t = 1, 2, 3, \\ 0.1R^*_{j,t-3} + 0.2R^*_{j,t-2} + 0.3R^*_{j,t-1} + 0.4\widetilde{R}_{j,t} & t > 3. \end{cases} \tag{2.10}$$

Dalla (2.10) è chiaro che per le prime tre partite di campionato il punteggio $\widetilde{R}_j$ non verrà modificato a differenza delle successive. Una volta stimato $R^*_{j,4}$, questo verrà utilizzato per la stima di $R^*_{j,5}$ con il quale a sua volta si stimerà $R^*_{j,6}$ e così via.

Nella giornata t-esima per il calcolo del *ranking* finale viene dunque utilizzato il 40% del punteggio stimato precedentemente per quella partita e il 60% del punteggio di giornate precedenti. Questi valori sembrano sensati considerando che si vuole sia essere sensibili a cambiamenti di stato di forma dei giocatori di giornata in giornata, sia avere una buona memoria delle partite precedenti.

Come esemplificazione, in Figura 2.4 è possibile osservare l'andamento di 4 diversi ottimi giocatori durante tutto il campionato e la rispettiva curva lisciata con la combinazioni di pesi (0.1, 0.2, 0.3, 0.4). Avendo a disposizione l'evoluzione del punteggio dei giocatori durante l'intera stagione, è interessante notare come sia possibile effettuare delle valutazioni mirate a piccoli periodi del campionato, come l'andamento altalenante di Nikola Jokic o la lenta e costante crescita di James Harden.

Player	**Score**	**Match**	**Pos**	**Team**
Joel Embiid	0.8564	75	C	PHI
Giannis Antetokounmpo	0.8425	87	PF	MIL
Anthony Davis	0.7595	56	C	NOP
James Harden	0.7424	89	PG	HOU
Jonas Valanciunas	0.7254	49	C	TOR
Kawhi Leonard	0.6779	84	SF	TOR
Nikola Jokic	0.6693	94	C	DEN
Julius Randle	0.6680	73	PF	NOP
Jusuf Nurkic	0.6661	72	C	POR
Kevin Durant	0.6657	90	SF	GSW
Andre Drummond	0.6638	83	C	DET
Russell Westbrook	0.6581	78	PG	OKC
Karl-Anthony Towns	0.6557	77	C	MIN
LeBron James	0.6506	55	SF	LAL
Paul George	0.6418	82	SF	OKC

Tabella 2.7: Classifica finale dei 15 migliori giocatori di tutto il campionato

2.3 Classifica finale

Si hanno ora a disposizione tutti gli ingredienti per valutare le singole *performance* degli atleti. L'aggiornamento dello *score* descritto nei precedenti paragrafi permette di valutare in modo dinamico lo stato di forma dei giocatori.

Per tenere conto sia del numero di partite svolte fino alla giornata t, sia della presenza di alcuni *outlier*, si è deciso di calcolare il punteggio finale come mediana dei punteggi fino al tempo t e di considerare solo i giocatori con almeno il 40% di incontri del giocatore con più incontri della lega, un numero molto basso considerando che i *top player* arrivano a disputare anche 100 partite a fine stagione, escludendo così pochi atleti. La mediana assegna stesso peso a tutte le osservazioni fino all'istante considerato: questo approccio è sensato dato che la classifica finale deve tenere conto dell'evoluzione delle prestazioni fin dal primo giorno e non limitarsi unicamente alle giornate recenti.

L'operazione ha portato alla classifica di fine campionato visibile in Tabella 2.7. Tra le variabili presenti, oltre al giocatore e al rispettivo *score* stimato, compare anche il numero di partite disputate, la posizione e la squadra di appartenenza. È interessante notare come il numero di incontri giocati sia sì una variabile rilevante, dato che molti dei giocatori selezionati hanno più di 80 incontri, ma non sia l'unica discriminante.

Altro aspetto interessante è il ruolo dei giocatori: nonostante prevalga il "Centrale", si nota una certa variabilità di posizione. Questo è dovuto alla natura della variabile

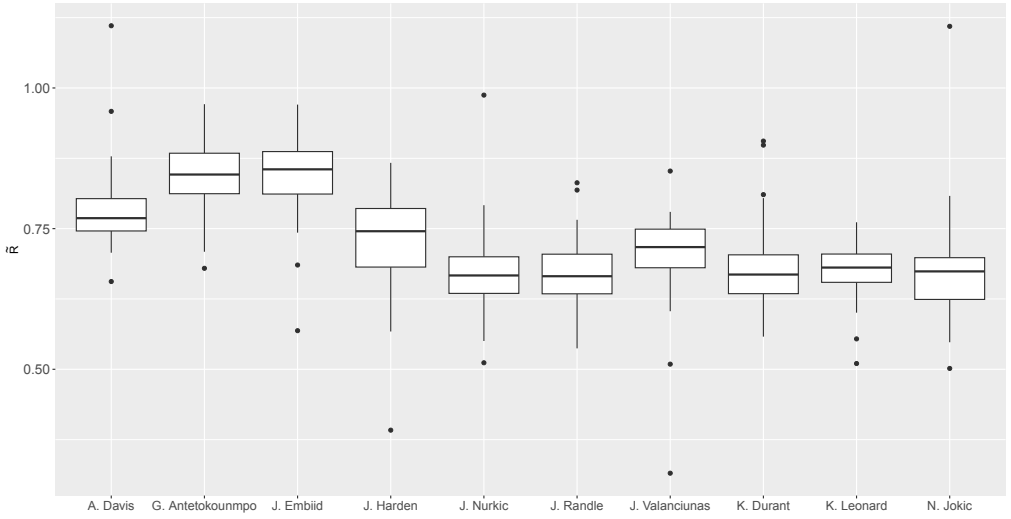

FIGURA 2.5: Distribuzione dei punteggi $\widetilde{R}$ a fine campionato per i primi 10 giocatori

risposta, poiché un evento viene considerato pericoloso se è stato di contributo al canestro. Di conseguenza, anche giocatori non addetti al tiro al canestro o alla schiacciata possono risultare fondamentali nella realizzazione dell'azione.

Al primo posto della classifica si trova Joel Embiid, giocatore di punta dei Philadelphia 76ers, incluso, nella stagione considerata, sia nel secondo quintetto All-NBA che nel secondo quintetto All-Defense. Quell'anno ha inoltre stabilito *record* personali ancora imbattuti, come 13.6 rimbalzi in media a partita.

Al secondo posto si trova Giannis Antetokounmpo, eletto MVP della Regular Season 2018/19 e protagonista di ben 60 vittorie in campionato. Non sorprende che i primi posti della classifica siano occupati da questi due giocatori.

Il terzo posto è occupato da Anthony Davis, giocatore molto versatile, il quale quell'anno ricopre il ruolo di uno dei migliori difensori del campionato.

Con 36.1 punti di media James Harden si aggiudica il quarto posto, eletto inoltre miglior giocatore in termini di BPM (Tabella 1.5) della stagione. Le sue incredibili prestazioni lo portano ad occupare uno dei gradini più alti della classifica.

I due giocatori successivi appartengono entrambi ai Toronto Raptors, squadra vincitrice del titolo. In particolare, Kawhi Leonard viene nominato MVP delle Finals per la seconda volta in carriera, diventando il terzo giocatore della storia a vincere il premio con due squadre diverse. Jonas Valanciunas viene invece ceduto ai Memphis Grizzlies prima dell'inizio dei Playoff, firmando solamente 30 presenze nei Toronto Raptors.

In generale, tutti i giocatori selezionati possono vantare un'ottima stagione, molti dei quali sono stati inseriti nei migliori quintetti NBA, suggerendo che il procedimento adottato sembrerebbe sensato. In Figura 2.5 è possibile osservare come varia la distribuzione

degli $\widetilde{R}$ per i primi 10 giocatori selezionati. I *top* 5 *player* sembrerebbero distaccarsi nettamente rispetto agli altri giocatori, ad eccezione di James Harden per il quale risulta una distribuzione dei punteggi più variabile. In generale, il grafico suggerisce che la mediana risulta una buona statistica per riassumere le *performance* dei giocatori a fine campionato.

Si noti come alcuni giocatori selezionati dall'indice BPM non compaiano nella classifica appena presentata. L'indice, infatti, individua il contributo in termini di incremento di punti segnati quando il giocatore è in campo, leggermente diverso dalla variabile risposta creata nella presente tesi, la quale tiene conto del contributo del giocatore nel segnare il canestro, mirato dunque a valutare la pericolosità offensiva. Di conseguenza delle leggere differenze sono giustificate.

Si ritiene, in conclusione, di valutare i risultati come affidabili e pertanto proseguire con ulteriori analisi.

2.3.1 Classifica per ruolo

Una volta stilata una classifica generale dei giocatori è possibile svolgere alcune analisi mirate, ad esempio stratificando per ruolo. In questo modo si riescono a confrontare giocatori appartenenti alla stessa posizione. Le classifiche stratificate sono disponibili nelle Tabelle 2.8, 2.9, 2.10, 2.11 e 2.12. Quello che salta subito all'occhio è la comparsa di ottimi giocatori nelle posizioni più alte i quali, nella classifica generale, erano stati esclusi, come Stephen Curry e Kyrie Irving, presenti nella classifica secondo BPM visibile in Tabella 1.5. In particolare questo suggerirebbe l'importanza di considerare tale diversificazione per l'analisi in esame, poiché alcuni ruoli presentano generalmente punteggi più alti di altri, come il "Centrale", osservando la Figura 2.6.

Player	**Score**	**Match**	**Team**
James Harden	0.7424	89	HOU
Russell Westbrook	0.6581	78	OKC
Damian Lillard	0.5710	96	POR
Stephen Curry	0.5673	91	GSW
Kyrie Irving	0.5513	76	BOS
Kemba Walker	0.5388	82	CHA
Ben Simmons	0.5235	91	PHI
Trae Young	0.5026	81	ATL
Mike Conley	0.4993	70	MEM
De'Aaron Fox	0.4932	81	SAC

Tabella 2.8: Classifica dei migliori 10 "Play maker"

Player	Score	Match	Team
Devin Booker	0.6181	64	PHX
Lou Williams	0.6056	81	LAC
Zach LaVine	0.5857	63	CHI
DeMar DeRozan	0.5607	84	SAS
Donovan Mitchell	0.5328	82	UTA
Bradley Beal	0.5265	82	WAS
Luka Doncic	0.5235	72	DAL
Jimmy Butler	0.5191	77	PHI
Jordan Clarkson	0.5039	81	CLE
Tim Hardaway Jr.	0.4842	65	NYK

Tabella 2.9: Classifica delle migliori 10 "Guardie tiratrici"

Player	Score	Match	Team
Kawhi Leonard	0.6779	84	TOR
Kevin Durant	0.6657	90	GSW
LeBron James	0.6506	55	LAL
Paul George	0.6418	82	OKC
Danilo Gallinari	0.5422	74	LAC
Khris Middleton	0.5017	92	MIL
Brandon Ingram	0.4778	52	LAL
Jayson Tatum	0.4713	88	BOS
Caris LeVert	0.4684	45	BKN
Rondae Hollis-Jefferson	0.4542	60	BKN

Tabella 2.10: Classifica delle migliori 10 "Ali piccoli"

Player	Score	Match	Team
Giannis Antetokounmpo	0.8425	87	MIL
Julius Randle	0.6680	73	NOP
Marvin Bagley III	0.6271	62	SAC
Blake Griffin	0.6236	77	DET
John Collins	0.6130	61	ATL
Bobby Portis	0.5184	50	WAS
Paul Millsap	0.5132	84	DEN
Lauri Markkanen	0.5004	52	CHI
Jaren Jackson Jr.	0.4890	58	MEM
Cheick Diallo	0.4866	61	NOP

Tabella 2.11: Classifica delle migliori 10 "Ali grandi"

Player	Score	Match	Team
Joel Embiid	0.8564	75	PHI
Anthony Davis	0.7595	56	NOP
Jonas Valanciunas	0.7254	49	TOR
Nikola Jokic	0.6693	94	DEN
Jusuf Nurkic	0.6661	72	POR
Andre Drummond	0.6638	83	DET
Karl-Anthony Towns	0.6557	77	MIN
Nikola Vucevic	0.6320	85	ORL
Hassan Whiteside	0.6301	72	MIA
Montrezl Harrell	0.6184	88	LAC

Tabella 2.12: Classifica dei migliori 10 "Centrali"

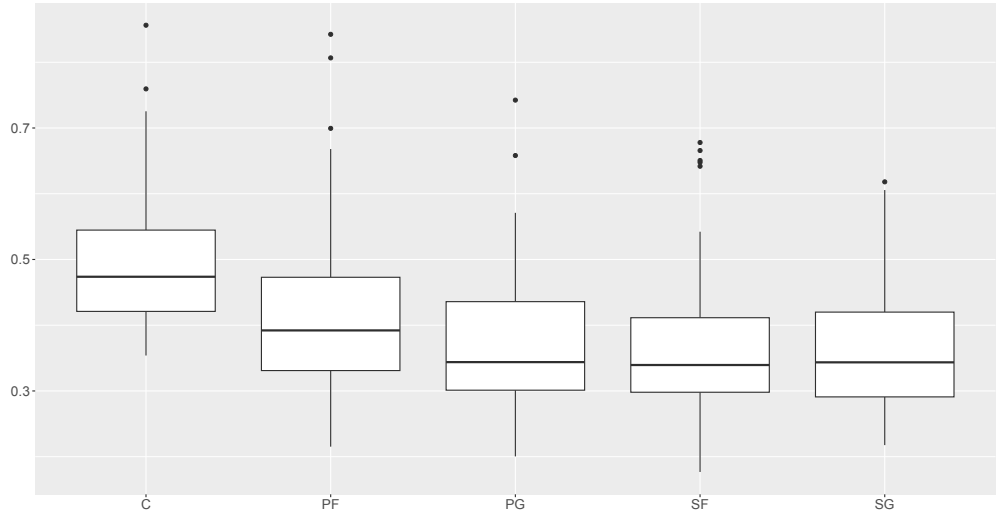

Figura 2.6: Distribuzione dei punteggi finali a fine campionato per ruolo

2.3.2 Classifica per squadra

Un'altra analisi possibile riguarda una valutazione delle *performance* delle squadre. Avendo a disposizione i punteggi dei singoli giocatori, una possibilità è quella di aggregare i risultati degli atleti stratificando per la squadra in cui hanno giocato. Bisogna prestare però particolare attenzione a quei giocatori che durante la stagione hanno giocato in più di una squadra. Non è quindi così immediato eseguire un confronto di squadra.

Si è deciso dunque di considerare, come punteggio di squadra, la media di tutti i singoli punteggi dei giocatori appartenenti a essa. Questo ha richiesto un ricalcolo di tutti i punteggi degli atleti dato che per l'attribuzione del punteggio finale proposto nel

Team	Score
OKC	0.43411
DEN	0.43410
PHI	0.43402
TOR	0.42832
ATL	0.42661
SAC	0.42635
MIL	0.42284
LAC	0.42283
UTA	0.41611
GSW	0.41493

Tabella 2.13: Classifica delle prime 10 squadre

Paragrafo 2.3 non era previsto un condizionamento alla squadra. Ogni giocatore avrà, dopo questa operazione, un numero totale di punteggi pari al numero di squadre in cui ha giocato. In sostanza si avrà

$$\hat{S}_s = \frac{1}{p_s} \sum_{j=1}^{p_s} \frac{1}{m_{js}} \sum_{t=1}^{m_{js}} R^*_{j,t}, \tag{2.11}$$

dove m_{js} indica il numero di partite disputate dal giocatore j nella squadra s e p_s è il numero di giocatori che hanno giocato per la squadra s. Per ogni squadra s si propone di fatto come punteggio una media di tutti i punteggi medi dei giocatori. In Tabella 2.13 è possibile vedere il risultato di questa operazione.
Ad occupare la prima posizione si trovano gli Oklahoma Thunder, squadra con la percentuale di palle rubate per gara più elevata, seguita dai Denver Nuggets, una delle migliori squadre a livello difensivo (BasketballReference, 2018), uscita ai quarti di finale ai Playoff. Al terzo posto si trovano i Philadelphia 76ers, squadra di Joel Embiid, non sorprende che sia una delle squadre nelle prime posizioni data la sua efficacia in attacco. Al quarto posto si trovano i Toronto Raptors, vincitori del titolo nell'annata 2018/19.

Ad eccezione degli Atlanta Hawks e dei Sacramento Kings, le altre squadre sono tutte arrivate in fase di Playoff, confermando dunque una buona stagione generale.

La classifica presentata, nonostante includa la maggior parte delle migliori squadre dell'annata considerata, non sembra essere in accordo con le più note statistiche disponibili. Probabilmente, il metodo proposto non è adatto per valutare le *performance* di squadra poiché è noto quest'ultime non sono una semplice somma di punteggi individuali, ma un vero e proprio lavoro di collaborazione tra atleti e allenatori. Pertanto questi risultati verranno presi con la giusta cautela. Interessante notare anche come la

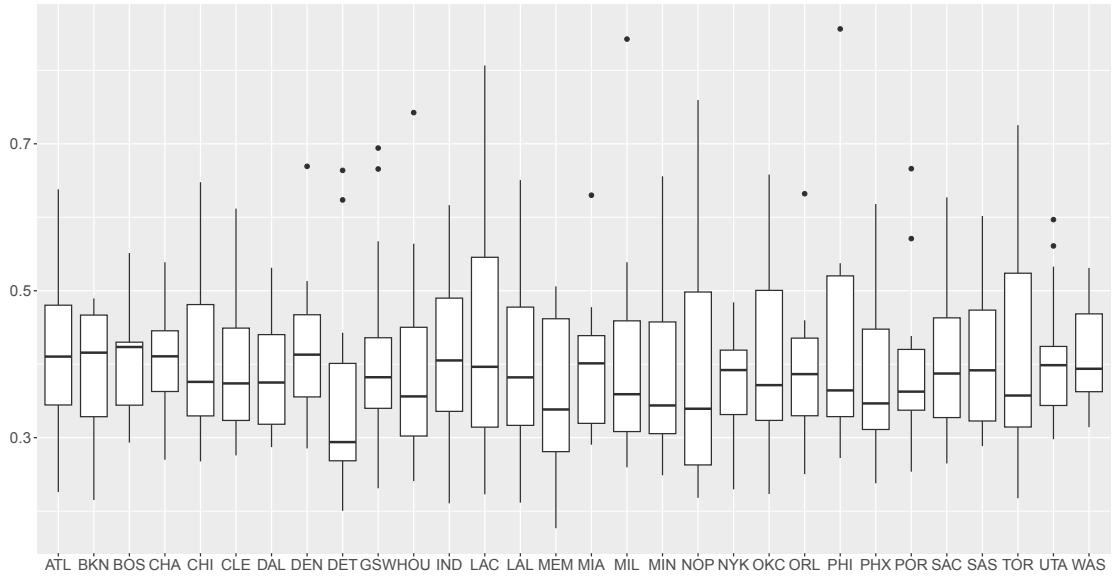

FIGURA 2.7: Distribuzione dei punteggi finali a fine campionato per squadra

distribuzione dei punteggi finali per squadra sia particolarmente eterogenea passando da una squadra all'altra, rendendo difficile arrivare a delle conclusioni certe, come chiaro dalla Figura 2.7.

2.3.3 Regular Season e Playoff

Come ultima analisi si vuole capire se esistono differenze in termini di classifica tra la Regular Season e i Playoff, dato che i due momenti distinti del campionato vengono notoriamente trattati in modo separato nella creazione delle statistiche.

Condizionandosi ai due periodi e calcolando nuovamente la mediana dei punteggi, come visto nel Paragrafo 2.3, le due classifiche assumono la forma visibile in Tabella 2.14.

È possibile osservare come la prima posizione vari a seconda del periodo del campionato. Non sorprende che Giannis Antetokounmpo salga in prima posizione, poiché nella stagione considerata è proprio il detentore del titolo di MVP della Regular Season. Le sue prestazioni non sono però altrettanto buone nei Playoff, sebbene rimangano ottime.

Anthony Davis non compare invece nella classifica dei Playoff, il che non sorprende dato che i New Orleans Pelicans non sono arrivati quell'anno tra le 16 squadre finali. Interessante notare come Nikola Jokic guadagni invece diverse posizioni, firmano proprio nei Playoff di quell'annata il suo personale di tiri liberi messi a segno, pari a 84.6%. Si noti infine come anche Kawhi Leonard, vincitore del campionato e del titolo MVP dei Playoff, guadagni una posizione, concludendo in quinta posizione.

Regular Season	
Player	**Score**
Giannis Antetokounmpo	0.8614
Joel Embiid	0.8513
Anthony Davis	0.7595
James Harden	0.7520
Jonas Valanciunas	0.7254
Kawhi Leonard	0.6830
Andre Drummond	0.6801
Kevin Durant	0.6754
Nikola Jokic	0.6725
Julius Randle	0.6680
Jusuf Nurkic	0.6661
Russell Westbrook	0.6623
Karl-Anthony Towns	0.6557
LeBron James	0.6506
Paul George	0.6468

Playoff	
Player	**Score**
Joel Embiid	0.8122
Giannis Antetokounmpo	0.7394
Nikola Jokic	0.6944
James Harden	0.6724
Kawhi Leonard	0.6633
Kevin Durant	0.6402
Lou Williams	0.6242
DeMarcus Cousins	0.5925
Stephen Curry	0.5759
Montrezl Harrell	0.5738
Kyrie Irving	0.5519
DeMar DeRozan	0.5434
LaMarcus Aldridge	0.5419
Damian Lillard	0.5290
Greg Monroe	0.5202

Tabella 2.14: Classifica dei migliori 15 giocatori condizionata al periodo

Sembra dunque chiaro esserci differenza tra le due fasi del campionato, tuttavia è bene ricordare che le stime di probabilità per ogni giornata risentono delle stime delle giornate precedenti, come chiarito nel Paragrafo 2.2.3. Di conseguenza, specialmente per le prime giornate di Playoff, le stime di pericolosità degli atleti sono naturalmente influenzate dalle ultime giornate di Regular Season. Le conclusioni riguardo i Playoff devono dunque essere prese con la giusta attenzione, specialmente per quei giocatori con poche partite. Volendo ottenere delle classifiche isolate, si potrebbe ripetere l'intera analisi descritta nel presente capitolo senza considerare la Regular Season, creando però delle stime poco affidabili e basate su un basso numero di partite. In alternativa, si potrebbero modificare i pesi della media mobile descritta nel Paragrafo 2.2.3 in modo tale da diminuire l'influenza della Regular Season sui Playoff.

2.4 Valutazioni

L'approccio proposto nel presente capitolo si pone l'obiettivo di creare un *ranking* per giocatore aggiornato di partita in partita considerando le giornate indipendenti tra di loro e di stimare dei modelli separati su ogni singola giornata. Il punto debole del

metodo risiede proprio in questa assunzione, dato che non è realistico trascurare un ordinamento temporale.

Per ovviare a questo problema è stato proposto un aggiustamento dei punteggi per il numero di eventi al minuto e un lisciamento tramite media mobile, in modo da includere della dipendenza temporale tra le giornate di gioco.

I risultati delle analisi possono essere considerati soddisfacenti. La classifica generale individua come migliori giocatori quelli che hanno effettivamente avuto un'ottima annata. I punteggi per squadra non sembrano d'altra parte essere particolarmente affidabili, mentre il condizionamento al ruolo e al periodo risulta necessario, dato che emergono eterogeneità tra i vari gruppi.

Per il successivo capitolo ci si pone l'obiettivo di creare modelli, tramite un approccio bayesiano, che tengano conto dell'ordinamento temporale dei dati, aggiornando i parametri con il susseguirsi delle partite in maniera sequenziale, superando i limiti del metodo appena presentato.

Capitolo 3

Approccio bayesiano

Il presente capitolo si pone l'obiettivo di rispondere alla domanda d'interesse tramite approcci bayesiani. Lo scopo è quello di aggiornare i parametri del modello di regressione binaria man mano che sopraggiungono nuovi incontri, evitando di dover adattare dei modelli separati sui soli dati di ogni singola giornata. Verranno utilizzati diversi approcci in termini di metodo, di complessità e di costo computazionale. Una volta ottenute le stime di pericolosità degli atleti di giornata in giornata, si procederà a un'aggregazione dei valori come descritto nel Capitolo 2, ossia tramite aggiustamento per il numero di eventi al minuto e successiva applicazione della media mobile pesata.

3.1 Inferenza bayesiana

Nell'inferenza statistica classica o frequentista vengono usati modelli dove si assume che i parametri siano costanti non note. I dati campionari vengono quindi utilizzati per pervenire ad una stima (puntuale o di intervallo) o per sottoporre a verifica empirica ipotesi riguardanti tali parametri. Nell'approccio classico i dati campionari sono l'unica fonte utilizzata ed utilizzabile per giungere ad una conoscenza "oggettiva" della realtà rispetto alla quale non si presuppone alcuna conoscenza pregressa. Nell'approccio bayesiano, invece, una tale conoscenza si presuppone e i dati campionari servono solo per procedere al suo aggiornamento. Il problema si risolve considerando i parametri non più delle costanti incognite ma delle variabili casuali governate da una propria distribuzione di probabilità.

Supponendo dunque di essere interessati a un generico parametro $\theta \in \mathbb{R}^p$ si assume che questo sia realizzazione di una variabile casuale con distribuzione nello spazio parametrico $\Theta \in \mathbb{R}^p$, avente una densità a priori $\pi(\theta)$, la quale riassume le informazioni

preliminari su θ prima dell'osservazione dei dati. Una volta ricevuto il generico vettore di osservazioni $y = (y_1, ..., y_n)$ è possibile aggiornare la distribuzione di θ dato y servendosi del teorema di Bayes, ottenendo una nuova distribuzione, detta a posteriori. Formalmente si ha che

$$\pi(\theta|y) = \frac{f(y|\theta)\pi(\theta)}{f(y)} = \frac{f(y|\theta)\pi(\theta)}{\int_\Theta \pi(y|\theta)\pi(\theta)d\theta}, \tag{3.1}$$

dove nel caso di osservazioni indipendenti

$$f(y|\theta) = f(y_1, ..., y_n|\theta) = \prod_{i=1}^{n} f(y_i|\theta).$$

In questo caso $f(y|\theta)$ indica la densità congiunta del vettore y, proporzionale alla funzione di verosimiglianza $L(\theta)$. Il termine al denominatore della (3.1) può essere trattato come una costante moltiplicativa di normalizzazione dato che non dipende da θ, riducendosi a

$$\pi(\theta|y) \propto L(\theta)\pi(\theta). \tag{3.2}$$

Il problema principale di questo approccio risiede nel calcolo del denominatore appena citato, il quale non sempre può essere ricavato per via analitica, specialmente quando la dimensione dello spazio parametrico aumenta. Esistono vari metodi di simulazione utilizzabili in contesti in cui non è possibile ricavare una densità a posteriori nota, i quali tuttavia possono richiedere uno sforzo computazionale non indifferente.

Un'alternativa ai metodi di simulazione è l'utilizzo di densità a priori coniugate per i parametri, ossia funzioni che, una volta aggiornate tramite la verosimiglianza, appartengono alla stessa famiglia delle distribuzioni a priori: in questi casi, è necessario un semplice aggiornamento del valore dei parametri, una volta sopraggiunti i dati, operazione che non richiede l'utilizzo di simulazioni.

Altra strada percorribile è l'utilizzo di approssimazioni della distribuzione a posteriori che rendano più semplice una stima dei parametri.

Nell'ambito del presente studio è necessario tenere in considerazione questi aspetti, dato che la dimensione dello spazio parametrico è elevata. Verranno di conseguenza esplorate diverse alternative in modo da affrontare i problemi appena presentati.

3.2 Approssimazione normale

Una prima strada che si è voluta percorrere è l'utilizzo di una approssimazione per la distribuzione a posteriori basata sulla teoria asintotica.

Il risultato principale della teoria asintotica bayesiana è il teorema di Bernstein-von Mises, il quale afferma che, sotto opportune condizioni di regolarità, quando n è sufficientemente grande, la distribuzione a posteriori può essere approssimata da una distribuzione normale.

Le condizioni di regolarità richieste sono essenzialmente le stesse necessarie per la normalità asintotica dello stimatore di massima verosimiglianza. In particolare, l'approssimazione normale vale per distribuzioni a posteriori unimodali, il cui massimo è un punto interno di Θ, e per densità a priori strettamente positive su Θ e sufficientemente lisce. Per una trattazione approfondita si veda, ad esempio, Salvan et al. (2023).

Sotto queste condizioni di regolarità, indicando con $\tilde{\theta}$ la moda della distribuzione a posteriori e $\tilde{J}(\tilde{\theta})$ la derivata seconda della log-posteriori cambiata di segno e valutata in $\tilde{\theta}$, vale l'approssimazione asintotica

$$\theta|y \dot{\sim} N_p(\tilde{\theta}, \tilde{J}(\tilde{\theta})^{-1}),$$

ossia la distribuzione a posteriori può essere approssimata con una normale p-dimensionale. Logicamente all'aumentare di n, aumenta l'accuratezza di tale approssimazione.

Nel caso in esame il parametro di interesse θ è rappresentato dal vettore $\beta \in \mathbb{R}^p$, con $p = 470$.

L'attenzione si sposta ora sulla scelta di una priori che sia adeguata al caso di interesse. Una possibilità è quella di servirsi di una priori di facile utilizzo, come quella normale, essendo quest'ultima una scelta comune per parametri di regressione in modelli lineari generalizzati. Malgrado l'utilizzo della distribuzione a priori normale non includa possibili asimmetrie per la distribuzione dei parametri, si è inizialmente deciso di percorrere questa strada per la semplicità di implementazione.

Sia dunque β_0 il vettore di medie a priori, Σ_0 la matrice di varianze e covarianze a priori, X e y, ancora una volta, la matrice del disegno e il vettore relativo alla variabile risposta, si assume allora che

$$\beta \sim N_p(\beta_0, \Sigma_0), \qquad \beta|X, y \dot{\sim} N_p(\tilde{\beta}, \tilde{J}(\tilde{\beta})^{-1}).$$

Utilizzando la (3.2) è possibile mettere in relazione le due quantità appena descritte come

$$\pi(\beta|X,y) \propto L(\beta)\pi(\beta). \tag{3.3}$$

Per capire come procedere con l'aggiornamento dei parametri di interesse è necessario conoscere la natura di $L(\beta)$ e le quantità che ne derivano. Supponendo di utilizzare nuovamente il modello *probit* per il caso di regressione binaria in esame è chiaro che la funzione di verosimiglianza è esprimibile come descritto nella (2.2). La log-verosimiglianza avrà dunque una forma del tipo

$$l(\beta; X, y) = l(\beta) = \sum_{i=1}^{n} y_i \log(\Phi(\boldsymbol{x_i}^T\beta)) + (1-y_i)\log(1-\Phi(\boldsymbol{x_i}^T\beta)).$$

Derivando in β si ottiene la funzione punteggio

$$U(\beta) = \frac{\partial}{\partial\beta} l(\beta) = \sum_{i=1}^{n} \left(\frac{y_i}{\Phi(\boldsymbol{x_i}^T\beta)} + \frac{(y_i-1)}{1-\Phi(\boldsymbol{x_i}^T\beta)} \right) \phi(\boldsymbol{x_i}^T\beta)\boldsymbol{x_i}^T$$

e derivando nuovamente in β

$$\begin{aligned} J(\beta) = -\frac{\partial}{\partial\beta^T} U(\beta) = &\sum_{i=1}^{n} y_i \left(\frac{\phi(\boldsymbol{x_i}^T\beta)^2}{\Phi(\boldsymbol{x_i}^T\beta)^2} + \frac{\phi(\boldsymbol{x_i}^T\beta)\boldsymbol{x_i}^T\beta}{\Phi(\boldsymbol{x_i}^T\beta)} \right) \boldsymbol{x_i}\boldsymbol{x_i}^T + \\ &\sum_{i=1}^{n} (1-y_i) \left(\frac{\phi(\boldsymbol{x_i}^T\beta)^2}{(1-\Phi(\boldsymbol{x_i}^T\beta))^2} - \frac{\phi(\boldsymbol{x_i}^T\beta)\boldsymbol{x_i}^T\beta}{1-\Phi(\boldsymbol{x_i}^T\beta)} \right) \boldsymbol{x_i}\boldsymbol{x_i}^T, \end{aligned}$$

ricordando che $\phi(u)' = -u\phi(u)$.

Si hanno ora tutti gli ingredienti per procedere con l'aggiornamento dei parametri. Riprendendo la (3.3) in scala logaritmica si ha che

$$\tilde{l}(\beta) = -\frac{1}{2}(\beta-\beta_0)^T\Sigma_0^{-1}(\beta-\beta_0) + l(\beta),$$

e derivando nuovamente in β

$$\begin{aligned} \tilde{U}(\beta) &= \frac{\partial}{\partial\beta}\tilde{l}(\beta) = \Sigma_0^{-1}(\beta_0-\beta) + U(\beta), \\ \tilde{J}(\beta) &= -\frac{\partial}{\partial\beta^T}\tilde{U}(\beta) = \Sigma_0^{-1} + J(\beta). \end{aligned} \tag{3.4}$$

Dalla (3.4) è chiaro come l'aggiornamento della matrice di varianza e covarianza richieda una semplice somma tra matrici, divisa in contributo a priori (Σ_0^{-1}) e nuova informazione portata dalla verosimiglianza ($J(\beta)$).

Più delicato è invece l'aggiornamento della media della distribuzione a posteriori. La moda a posteriori $\tilde{\beta}$ risolve l'equazione

$$\tilde{U}(\beta) = 0_p$$

la quale non ammette soluzione esplicita, rendendo necessario l'utilizzo di metodi numerici. Svolgendo dunque uno sviluppo di Taylor arrestato al primo ordine intorno a $\tilde{\beta}$ partendo da un generico β_1 è noto che

$$\underbrace{\tilde{U}(\tilde{\beta})}_{=0_p} \simeq \tilde{U}(\beta_1) + \underbrace{\frac{\partial}{\partial\beta}\tilde{U}(\beta)\Big|_{\beta=\beta_1}}_{\tilde{J}(\beta_1)}(\tilde{\beta} - \beta_1).$$

Isolando $\tilde{\beta}$ si ottiene una prima stima del vettore di medie a posteriori, il quale verrà utilizzato come nuovo β_1 nell'iterazione successiva, così via fino a convergenza. Alla generica iterazione m-esima si avrà dunque una situazione del tipo

$$\tilde{\beta}_{m+1} \simeq \tilde{\beta}_m + \tilde{J}(\tilde{\beta}_m)^{-1}\tilde{U}(\tilde{\beta}_m).$$

Una possibile scelta come punto di partenza per l'approssimazione di $\tilde{\beta}$ è la media della distribuzione a priori β_0, poiché, se la priori è informativa, è ragionevole pensare che le due quantità siano vicine, riducendo così il numero di iterazioni.

È possibile ora presentare l'algoritmo con il quale si ottiene un aggiornamento sequenziale dei parametri.

3.2.1 Aggiornamento normale-normale

Una volta chiarite tutte le componenti necessarie all'aggiornamento dei parametri nel caso di un'approssimazione asintotica gaussiana è possibile descrivere l'algoritmo utilizzato.

Per la prima partita di campionato è necessario assegnare dei valori al vettore di medie β_0 e alla matrice Σ_0, dato che non si hanno informazioni a disposizione. Una possibilità è quella di utilizzare delle distribuzioni a priori poco informative, ossia che non diano forti informazioni preliminare sui parametri prima dell'arrivo dei dati. Altra possibilità prevede di utilizzare le prime giornate di gioco per ottenere delle stime, ad esempio con un semplice modello *probit*, da assegnare ai parametri della distribuzioni a priori. Questa strada è stata inizialmente scartata.

Seguendo la prima opzione, si è scelto di inizializzare β_0 con un vettore di 0, rimanendo quindi del tutto imparziali, mentre per Σ_0 una matrice diagonale con valori uguali a

Algoritmo 1 Aggiornamento normale-normale

Input: β_0, Σ_0, X, y, ε

for i in tutte le giornate **do**
  $m = 0, \quad d = (2\varepsilon, ..., 2\varepsilon)$ ▷ d vettore di differenze p-dimensionale
  while $any(d > \varepsilon)$ **do** ▷ Newton-Raphson
    $\beta_{m+1} = \beta_m + \tilde{J}(\beta_m)^{-1}\tilde{U}(\beta_m)$
    $d = |\beta_{m+1} - \beta_m|$
    $\beta_m = \beta_{m+1}$
    $m = m + 1$
  end while
  $\tilde{\beta}_i = \beta_m$
  $\beta_0 = \tilde{\beta}_i$ ▷ Aggiornamento delle nuove priori
  $\Sigma_0 = \tilde{J}(\tilde{\beta}_i)^{-1}$
end for

Output: $\tilde{\beta}_1, ..., \tilde{\beta}_n$

25, in modo tale da non considerare a priori nessun tipo di correlazione tra i parametri. Una varianza così alta non vincola inoltre i parametri ad uno stretto intervallo di valori.

Una volta scelta la distribuzione a priori, la procedura descritta nel Paragrafo 3.2 porta alla stima di un vettore $\tilde{\beta}$ e una matrice $\tilde{J}(\tilde{\beta})^{-1}$ a posteriori. Questi parametri verranno utilizzati nello *step* successivo come media e varianza della nuova distribuzione normale a priori, con lo scopo di includere, per la giornata successiva, l'informazione tratta dalla giornata di gioco appena conclusa. Così facendo col il passare delle giornate di gioco verrà inclusa informazione a priori basata su un numero crescente di dati.

Un altro importante ruolo della priori sta nel fatto di mantenere il valore dei parametri costanti quando gli eventi a cui questi si riferiscono non compaiono nella giornata: in particolare, a inizio campionato ogni parametro viene mantenuto a 0 finché non compare il relativo evento nel *dataset*. Si pensi, ad esempio, agli effetti fissi per i giocatori, i quali vengono inizializzati a 0 fino a che il giocatore a cui si riferiscono non è protagonista di un evento. In verità, una volta assunto un valore diverso da zero, sono visibili delle piccole variazioni del valore dei parametri, anche se il rispettivo evento non compare nella giornata, dovute probabilmente al popolarsi della matrici coinvolte nella stima.

Chiarito lo schema con cui verranno aggiornate le stime, in Algoritmo 1 è possibile osservare la procedura adottata con tutti i dati a disposizioni supponendo di ricevere il flusso di dati di giornata in giornata.

Una volta ottenuto, per ogni giornata di gioco, il vettore di mode a posteriori per il parametro β è possibile ricavare nuovamente le previsioni di pericolosità offensiva dei giocatori e procedere a stilare la classifica seguendo i passi del Capitolo 2. Tuttavia,

prima di presentare i risultati, è necessario soffermarsi su alcuni aspetti. Supponendo di trovarsi dunque alla t-esima giornata di campionato, l'aggiornamento della distribuzione a posteriori appena descritto avrà una forma del tipo

$$\pi(\beta|X^{(t)}, y^{(t)}) \propto \pi(\beta|X^{(t-1)}, y^{(t-1)})L(X^{(t)}, y^{(t)}|\beta),$$

dove $X^{(t)}$ e $y^{(t)}$ indicano i dati relativi alla t-esima giornata. Poiché la distribuzione a posteriori della giornata $(t-1)$-esima viene utilizzata come distribuzione a priori per la giornata successiva, la formula risulta chiara. In modo analogo è possibile eseguire delle sostituzioni all'indietro e ricondursi all'espressione compatta

$$\pi(\beta|X^{(t)}, y^{(t)}) \propto \pi(\beta) \prod_{i=1}^{t} L(X^{(i)}, y^{(i)}|\beta). \tag{3.5}$$

Dunque la distribuzione a posteriori per la giornata t-esima ingloba il contributo delle verosimiglianze calcolate per ogni giornata precedente, equamente pesate. Questo aspetto potrebbe risultare poco credibile, dato che ci si aspetta che giornate di gioco vicine a quella di interesse influenzino maggiormente le stime rispetto a quelle a inizio campionato. Si è voluto dunque tenere conto di questo aspetto e proporre una modifica del metodo appena presentato.

3.2.2 Modifica tramite *power prior*

Una possibilità per superare i limiti presentati nella parte conclusiva del paragrafo precedente è l'utilizzo della *power prior* (Ibrahim et al., 2015). Il metodo si basa sulla costruzione di una distribuzione a priori informativa proveniente da "dati storici", ossia, ad esempio, dati utilizzati in studi precedenti a quello in esame. Supponendo, ancora una volta, di essere interessati alla distribuzione di un generico parametro θ, di avere a disposizione dei dati storici y_0 e dei nuovi dati y, la distribuzione a posteriori per θ con l'aggiunta della *power prior* assume la forma

$$\pi(\theta|y, y_0, \alpha_0) \propto \pi_0(\theta)L(\theta|y)L(\theta|y_0)^{\alpha_0}, \tag{3.6}$$

dove $\pi_0(\theta)$ è la distribuzione a priori per θ prima di aver osservato y_0, mentre α_0 controlla l'influenza dei dati storici sulla distribuzione a posteriori. Tipicamente, il peso dei dati storici y_0 è minore rispetto al peso dei dati correnti y, il che suggerisce di far variare il parametro α_0 tra 0 e 1. Si noti che ponendo $\alpha_0 = 1$ ci si riconduce all'usuale aggiornamento bayesiano basato sui dati y e y_0, mentre $\alpha_0 = 0$ rappresenta la situazione di totale esclusione dell'informazione proveniente dai dati storici.

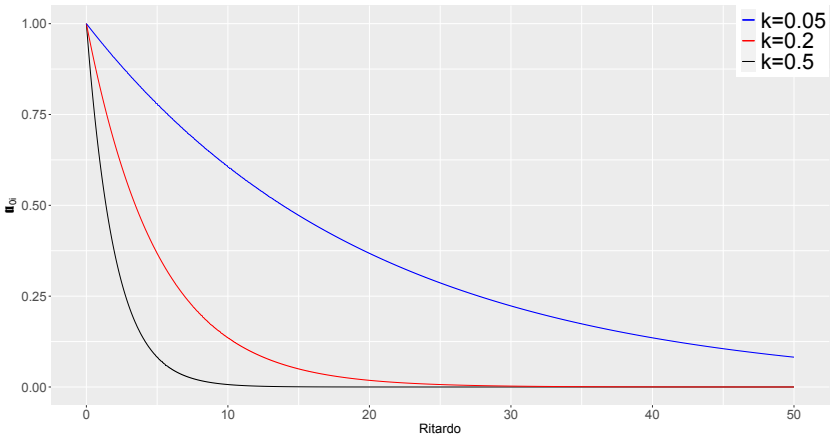

FIGURA 3.1: Peso α_{0i} all'aumentare della distanza tra le giornate

L'esempio può essere facilmente esteso in presenza di più di un *dataset* storico, come mostrato da Ibrahim et al. (2003). Supponendo dunque di essere in possesso di K_0 *dataset* storici, la (3.6) può essere generalizzata come

$$\pi(\theta|y, y_0, \alpha_0) \propto \pi_0(\theta)L(\theta|y)\prod_{k=1}^{K_0} L(\theta|y_{0k})^{\alpha_{0k}},$$

dove $0 \leq \alpha_{0k} \leq 1$, $k = 1, ..., K_0$. Il peso da assegnare ai parametri α_{0k} varia a seconda dell'importanza dei dati storici. Quest'ultima formulazione sembra essere utile per il caso di studio in esame. Riprendendo ora la (3.5), si è deciso di considerare come insieme di dati storici tutte le partite precedenti quella di interesse. In questo modo è possibile assegnare i pesi α_{0k} seguendo l'ordine temporale: partite distanti rispetto a quella corrente avranno peso minore rispetto a quelle recenti. Dunque la (3.5) viene modificata come

$$\pi(\beta|X^{(t)}, y^{(t)}) \propto \pi(\beta)L(X^{(t)}, y^{(t)}|\beta)\prod_{i=1}^{t-1} L(X^{(i)}, y^{(i)}|\beta)^{\alpha_{0i}}, \tag{3.7}$$

con $0 \leq \alpha_{01} \leq \alpha_{02} \leq ... \leq \alpha_{0t-1} \leq 1$. Si noti che assegnando tutti i pesi $\alpha_{0i} = 1, i = 1, ..., t-1$, ci si riconduce esattamente alla (3.5).

Per la scelta dei pesi da assegnare, si è deciso di utilizzare la relazione

$$\alpha_{0i} = e^{-k(t-i)},$$

con k parametro fissato. La quantità $t - i$ indica la distanza in giornate tra la partita corrente e la partita a cui voler assegnare il peso. La funzione esponenziale scelta risulta coerente con quanto discusso precedentemente, dato che all'aumentare della distanza tra le partite diminuisce il peso assegnato. In Figura 3.1 è possibile osservare come il

peso da assegnare ai dati precedenti dipende dal valore k.

Considerando il logaritmo della (3.7) e derivando rispetto a β si ottengono le nuove quantità

$$\tilde{l}^{(t)}(\beta) = -\frac{1}{2}(\beta - \beta_0)^T \Sigma_0^{-1}(\beta - \beta_0) + l(X^{(t)}, y^{(t)}|\beta) + \sum_{i=1}^{t-1} \alpha_{0i} l(X^{(i)}, y^{(i)}|\beta),$$

$$\tilde{U}^{(t)}(\beta) = \frac{\partial}{\partial \beta}\tilde{l}^{(t)}(\beta) = \Sigma_0^{-1}(\beta_0 - \beta) + U^{(t)}(\beta) + \sum_{i=1}^{t-1} \alpha_{0i} U^{(i)}(\beta),$$

$$\tilde{J}^{(t)}(\beta) = -\frac{\partial}{\partial \beta^T}\tilde{U}^{(t)}(\beta) = \Sigma_0^{-1} + J^{(t)}(\beta) + \sum_{i=1}^{t-1} \alpha_{0i} J^{(i)}(\beta),$$

dove $U^{(i)}(\beta)$ e $J^{(i)}(\beta)$ sono le quantità relative alla verosimiglianza della i-esima giornata. Dunque anche in questo caso il procedimento si riduce al solo aggiornamento della matrice $\tilde{J}^{(t)}(\beta)$ e del vettore $\tilde{U}^{(t)}(\beta)$, secondo le relazioni appena presentate. I passi successivi sono gli stessi dell'Algoritmo 1: ad ogni nuova giornata verrà calcolato, tramite l'algoritmo di Newton-Raphson, il vettore di mode a posteriori, il quale verrà utilizzato come media della distribuzione a priori per la giornata successiva, con la sola modifica del calcolo delle quantità $\tilde{J}^{(t)}(\beta)$ e $\tilde{U}^{(t)}(\beta)$ appena descritta. Tuttavia, in questo modo emergono dei problemi legati essenzialmente al carico computazionale.

Per la stima del vettore di mode a posteriori della giornata t-esima è necessario l'utilizzo dell'algoritmo di Newton-Raphson il quale, ad ogni iterazione, richiede il calcolo delle quantità $\tilde{U}^{(t)}(\beta)$ e $\tilde{J}^{(t)}(\beta)$. Le due quantità, a loro volta, necessitano del calcolo di $t-1$ verosimiglianze valutate in $t-1$ *dataset* diversi. Al crescere di t l'operazione può richiedere parecchio tempo. Per questo motivo si è scelto di valutare solamente un numero ristretto di giornate precedenti quella di interesse, pari a 7. Ciò che realmente influenza quanta informazione passata includere è la scelta del parametro k (Figura 3.1). Dopo aver esplorato diversi valori del parametro, vengono qui riportati i risultati ottenuti con l'utilizzo di $k = 0.5$ e $k = 0.05$, ossia due situazioni molto diverse tra loro osservando come diminuiscono i pesi all'aumentare del ritardo (Tabella 3.1).

Per ovviare al problema del carico computazionale si è sfruttato il pacchetto `OpenBLAS` (Xianyi et al., 2023), sviluppato per ottimizzare grandi calcoli matriciali. Tuttavia, i metodi appena presentati utilizzano matrici di dimensione massima $p \times p$. Con $p = 470$ e tramite l'utilizzo di questo pacchetto non si nota una riduzione del tempo di calcolo dato che, come anticipato, `OpenBLAS` risulta vantaggioso in presenza di matrici di dimensioni molto più grandi. Il caso in esame richiede invece molte operazioni tra matrici e vettori di dimensione moderata. Si è deciso dunque di non includere gli effetti fissi per giocatore

	k	
lag	0.5	0.05
1	0.61	0.95
2	0.37	0.90
3	0.22	0.86
4	0.14	0.82
5	0.08	0.78
6	0.05	0.74
7	0.03	0.70

TABELLA 3.1: Valore dei pesi α_{0i} al variare di k e del ritardo

e di inserire degli effetti fissi per squadra, in modo tale da ridurre la dimensione dello spazio parametrico includendo comunque informazione aggiuntiva oltre ai singoli eventi e ai ruoli dei giocatori, passando a $p = 76$. Questa semplificazione è necessaria se si vuole calcolare la classifica dal primo all'ultimo giorno di gara a causa dell'elevato carico computazionale. D'altra parte, supponendo di trovarsi durante il campionato e di dover aggiornare le stime con la sola partita corrente, l'operazione non richiederebbe più di qualche minuto includendo gli effetti fissi per giocatore, rendendo il procedimento fattibile.

In Figura 3.2 è possibile osservare l'andamento del valore di quattro diversi parametri rappresentanti gli eventi `Shot`, `foul`, `free_throw` e `steal` durante tutto il campionato. Sono stati utilizzati i valori del parametro k pari a 0.5 e 0.05 ed è stato inoltre adattato il modello privo della *power prior* descritto nel Paragrafo 3.2.1. Notando alla linea rossa e alla linea azzurra, ossia quelle relative ai modelli con *power prior*, si vede un certo accordo nell'evoluzione dei valori dei parametri. Entrambe le curve mostrano un'evoluzione nel tempo, il che non sorprende ricordando che per ogni giornata di gioco viene utilizzata solo una piccola parte del *dataset*, permettendo dunque alla curva finale di avere delle variazioni locali.

Discorso diverso vale invece per il modello privo dei pesi, rappresentato dalla linea nera. Per tutti e quattro i parametri presentati la curva sembrerebbe stabilizzarsi dopo poche partite. Anche questo risultato non sorprende, poiché andando avanti con il campionato la distribuzione a priori assume un peso sempre più di rilievo e l'influenza dei nuovi dati perde gradualmente di importanza. Dunque si nota una sorta di stabilizzazione delle stime. Si osservi, inoltre, che per come è definita la (3.5) l'aggiornamento della distribuzione a posteriori al tempo t risente dell'influenza di t verosimiglianze equamente pesate, dunque al crescere di t i dati relativi alla t-esima giornata hanno una piccola

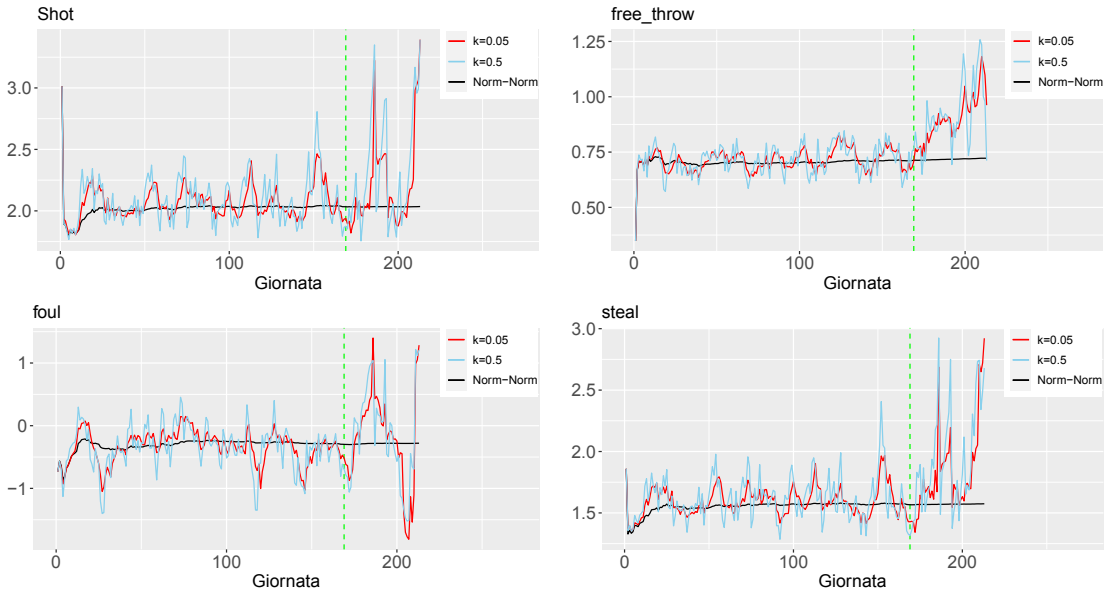

FIGURA 3.2: Andamento dei parametri `Shot`, `free_throw`, `foul` e `steal` per il modello senza utilizzo della *power prior*, il modello con *power prior* e $k = 0.5$ e il modello con *power prior* e $k = 0.05$

influenza sul totale, giustificando appunto l'appiattimento.

È stata inoltre aggiunta una linea verde verticale in corrispondenza della prima partita dei Playoff. È interessante notare come, rispetto alla Regular Season, le stime derivanti dai modelli a *power prior* siano molto più variabili suggerendo che, come già discusso nel Paragrafo 2.3.3, delle differenze tra i due momenti della stagione siano presenti. Bisogna però sottolineare che durante i Playoff vengono disputate meno partite rispetto alla Regular Season, il che suggerisce di interpretare i risultati con la giusta cautela.

3.2.3 Considerazioni e risultati

Una volta ottenuto il vettore di parametri β per ogni giornata di gioco è possibile ricavare una stima della variabile risposta e l'indice di pericolosità dei giocatori, come descritto nel Capitolo 2. Dopo aver calcolato, di giornata in giornata, il vettore di previsioni della variabile risposta, il primo passo dunque prevede di calcolarne il valore medio per giocatore, come descritto nel Paragrafo 2.2.1. Prima di procedere con le successive operazioni, in Figura 3.3 è possibile osservare un risultato particolarmente interessante. I tre diversi approcci messi a confronto nel paragrafo precedente, nonostante presentino valori dei coefficienti diversi con l'avanzare del campionato (Figura 3.2), sembrerebbero fornire delle prime stime dell'offensività dei giocatori simili. Questo risultato è coerente

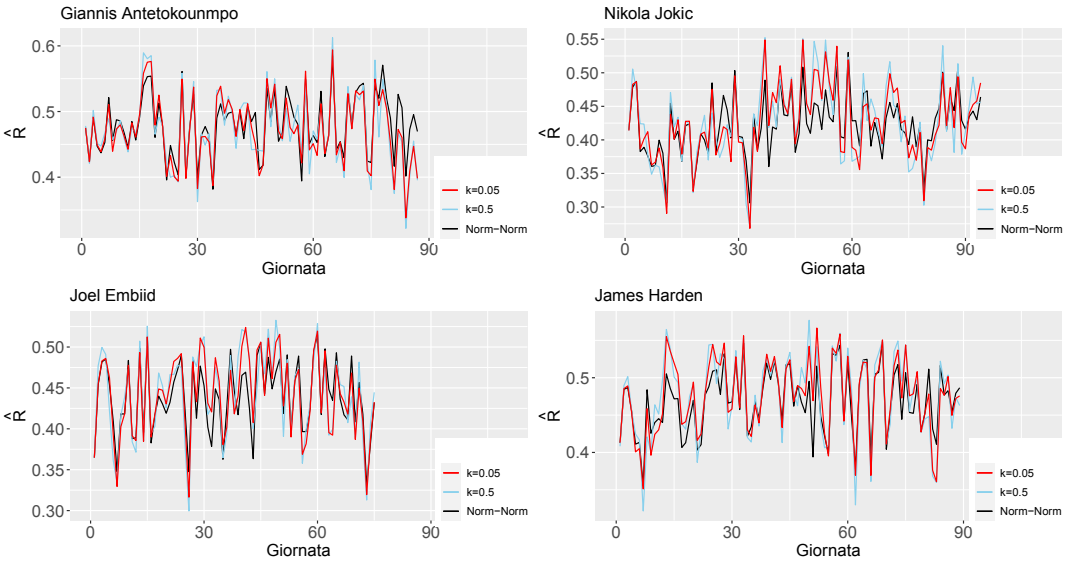

FIGURA 3.3: Andamento del coefficiente $\hat{R}$ per 4 giocatori per il modello senza utilizzo della *power prior*, il modello con *power prior* e $k = 0.5$ e il modello con *power prior* e $k = 0.05$

con la classifica finale visibile in Tabella 3.2, ottenuta dopo aver applicato l'aggiustamento per il coefficiente eventi al minuto (Paragrafo 2.2.2), il lisciamento tramite media mobile pesata (Paragrafo 2.2.3) e il calcolo della mediana (Paragrafo 2.3).

normale-normale		***k*=0.5**		***k*=0.05**	
Player	**Score**	**Player**	**Score**	**Player**	**Score**
G. Antetokounmpo	0.856	G. Antetokounmpo	0.845	G. Antetokounmpo	0.846
J. Embiid	0.829	J. Embiid	0.840	J. Embiid	0.840
A. Davis	0.784	A. Davis	0.768	A. Davis	0.767
J. Harden	0.740	J. Harden	0.745	J. Harden	0.745
J. Randle	0.699	J. Randle	0.699	A. Drummond	0.700
J. Valanciunas	0.685	A. Drummond	0.697	J. Randle	0.699
A. Drummond	0.684	J. Valanciunas	0.689	J. Valanciunas	0.680
L. James	0.670	J. Nurkic	0.668	J. Nurkic	0.666
J. Nurkic	0.664	R. Westbrook	0.665	L. James	0.664
K. Leonard	0.662	L. James	0.665	R. Westbrook	0.662

TABELLA 3.2: Classifica finale data dal modello senza utilizzo della *power prior*, il modello con *power prior* e $k = 0.5$ e il modello con *power prior* e $k = 0.05$

I tre metodi, infatti, sembrano essere in accordo su molte posizioni della classifica, presentando solamente alcune differenze dalla quinta posizione in giù. Anche il valore

finale dello *score* è estremamente simile tra i tre metodi (Tabella 3.2). Queste somiglianze suggeriscono che l'applicazione della *power prior* non sembra stravolgere la classifica ottenuta con il metodo privo dell'utilizzo dei pesi per le verosimiglianze, nonostante gli andamenti dei valori dei parametri siano diversi. Questo aspetto è da una parte rassicurante, poiché i migliori giocatori assoluti scelti, ad eccezione di qualche variazione, non sembrerebbero dipendere dal tipo di approccio utilizzato. D'altra parte, non è scontato capire come mai le medie di probabilità per giocatore siano così simili tra i metodi con e senza l'utilizzo della *power prior*. Una possibile spiegazione si basa sull'idea che, nonostante i singoli parametri varino durante il campionato, globalmente il predittore lineare restituisce delle probabilità analoghe a quelle del modello senza *power prior* e, una volta applicata la media per giocatore, le differenze tra i metodi si riducono. Questo aspetto risulta uno dei più interessanti della tesi e pone le basi per possibili sviluppi futuri.

3.3 Distribuzione *SUN*

Concluso un primo approccio tramite l'utilizzo dell'approssimazione asintotica della distribuzione a posteriori si vuole ora cercare un'alternativa che superi i limiti del metodo appena presentato. L'utilizzo di una distribuzione a priori normale può essere alle volte un'ipotesi restrittiva, poiché non si considera qualsiasi tipo di asimmetria per la distribuzione dei parametri. Inoltre l'approssimazione asintotica potrebbe essere poco adatta quando non si ha un buon numero di osservazioni a disposizione, dunque per le prime giornate di campionato. Questi aspetti hanno motivato la ricerca di un approccio diverso.

Prima di presentare il metodo utilizzato è necessario introdurre alcuni concetti. Sia dunque $z \sim SN_p(\xi, \Omega, \alpha)$ una normale asimmetrica multivariata (Azzalini & Dalla Valle, 1996) con distribuzione di probabilità $2\phi_p(z-\xi;\Omega)\Phi\{\alpha^T\omega^{-1}(z-\xi)\}$, ottenuta modificando quella di una normale p-dimensionale $N_p(\xi,\Omega)$ con l'aggiunta della funzione di ripartizione della normale standard calcolata in $\alpha^T\omega^{-1}(z-\xi)$, dove ω è una matrice diagonale $p \times p$ contenente le radici quadrate della diagonale di Ω. Questa strategia introduce asimmetria nella normale p-variata controllata da $\alpha = (\alpha_1, ..., \alpha_p)^T$, mentre il vettore $\xi = (\xi_1, ..., \xi_p)^T$ e la matrice Ω sono responsabili, rispettivamente, della posizione e della scala (Arellano-Valle & Azzalini, 2006). Si noti come ponendo $\alpha = 0_p$ si ottenga la $N_p(\xi, \Omega)$.

Motivati dal successo della formulazione sopra menzionata sono state proposte diverse estensioni per descrivere ulteriori proprietà. Due generalizzazioni importanti sono

ottenute aggiungendo un altro parametro γ a $\Phi\{\alpha^T\omega^{-1}(z-\xi)\}$ e consentendo al meccanismo responsabile dell'asimmetria di essere multivariato, incorporando una matrice Δ di dimensione $p \times n$ e una matrice di scala $n \times n$ Γ in $\Phi_n(\cdot)$. Grazie al contributo di Arellano-Valle & Azzalini (2006), queste generalizzazioni sono state unificate in un'unica distribuzione, detta normale asimmetrica unificata, con funzione di densità

$$\phi_p(z-\xi;\Omega)\frac{\Phi_n\{\gamma+\Delta^T\bar{\Omega}^{-1}\omega^{-1}(z-\xi);\Gamma-\Delta^T\bar{\Omega}^{-1}\Delta\}}{\Phi_n(\gamma;\Gamma)}, \tag{3.8}$$

dove ora $z \sim SUN_{p,n}(\xi,\Omega,\Delta,\gamma,\Gamma)$. In questo caso $\bar{\Omega}$ indica la matrice di correlazione ottenuta come $\omega^{-1}\Omega\omega^{-1}$. Le funzioni $\Phi_n(\cdot)$ presenti al numeratore e al denominatore indicano rispettivamente la funzione di ripartizione di una normale multivariata $N_n(0_n,\Gamma-\Delta^T\bar{\Omega}^{-1}\Delta)$ calcolata in $\gamma+\Delta^T\bar{\Omega}^{-1}\omega^{-1}(z-\xi)$ e la funzione di ripartizione di una $N_n(0_n;\Gamma)$ valutata in γ. Il vettore γ introduce ulteriore flessibilità e allontanamento dalla normalità mentre il principale effetto di asimmetria è dovuto alla matrice Δ.

Esistono ulteriori estensioni e modifiche della distribuzione normale asimmetrica le quali non verranno trattate nella seguente tesi in quanto non utili per l'obiettivo. Per approfondimenti si veda, ad esempio, Azzalini & Capitanio (2014).

3.3.1 *SUN* coniugata

Presentata la distribuzione SUN è possibile ora capire come verrà utilizzata in ottica *data stream*. Dal recente contributo di Durante (2019) è emerso che la SUN è la distribuzione coniugata per il modello *probit*. Questo risultato appare estremamente interessante per il caso in esame, permettendo, una volta che sopraggiungono i nuovi dati, di aggiornare i parametri della distribuzione a priori senza ricorrere ad approssimazioni o all'utilizzo di simulazioni.

Supponendo nuovamente di essere in un caso di regressione binaria con *link probit* e di indicare nuovamente con X la matrice del disegno e y il vettore relativo alla risposta e di aver definito una priori $SUN_{p,m}(\xi,\Omega,\Delta,\gamma,\Gamma)$, sfruttando ancora una volta la (3.2) e i risultati di Durante (2019) emerge che

$$\beta|X,y \sim SUN_{p,n+m}(\xi_{\text{post}},\Omega_{\text{post}},\Delta_{\text{post}},\gamma_{\text{post}},\Gamma_{\text{post}}),$$

con parametri d'aggiornamento $\xi_{\text{post}}=\xi$, $\Omega_{\text{post}}=\Omega$, $\Delta_{\text{post}}=(\Delta,\bar{\Omega}\omega D^T s^{-1})$, $\gamma_{\text{post}}=(\gamma^T,\xi^T D^T s^{-1})^T$ e Γ_{post} matrice di correlazione a rango pieno definita a blocchi, con $\Gamma_{\text{post[1,1]}}=\Gamma$, $\Gamma_{\text{post[2,2]}}=s^{-1}(D\Omega D^T+I_n)s^{-1}$ e $\Gamma_{\text{post[2,1]}}=\Gamma^T_{\text{post[1,2]}}=s^{-1}D\omega\Delta$. Inoltre D rappresenta la matrice $n \times p$ con $D=\text{diag}\{(2y_1-1),...,(2y_n-1)\}X$ e s la matrice

diagonale positiva $s = \text{diag}\{(d_1^T \Omega d_1 + 1)^{0.5}, ..., (d_n^T \Omega d_n + 1)^{0.5}\}$, dove d_i^T indica la i-esima riga di D.

Avendo a disposizione la distribuzione a posteriori per il parametro di interesse β l'attenzione si sposta ora su come scegliere un valore di quest'ultimo una volta ottenuta la distribuzione SUN a posteriori. Sfruttando i risultati di Azzalini & Bacchieri (2010) una possibilità è quella di utilizzare la media a posteriori, disponibile in forma chiusa, la quale non necessita l'utilizzo di simulazioni. In particolare si ha

$$\mathbb{E}_\beta(\beta|X, y) = \xi + \frac{\psi \omega \Delta_{\text{post}}}{\Phi_{m+n}(\gamma_{\text{post}}; \Gamma_{\text{post}})}, \tag{3.9}$$

dove ψ è un vettore $(n + m) \times 1$ con componente i-esima pari a

$$\psi_i = \phi(\bar{\gamma}_i)\Phi_{n+m-1}(\bar{\gamma}_{-i} - \bar{\Gamma}_{-i}\bar{\gamma}_i; \bar{\Gamma}_{-i,-i} - \bar{\Gamma}_{-i}\bar{\Gamma}_{-i}^T),$$

in cui $\bar{\gamma}_i$ e $\bar{\gamma}_{-i}$ denotano rispettivamente la i-esima componente di γ_{post} e il vettore $(n+m-1)\times 1$ ottenuto rimuovendo l'i-esima riga di γ_{post}. Con un ragionamento analogo, $\bar{\Gamma}_{-i,-i}$ indica la matrice ottenuta rimuovendo la i-esima riga e la i-esima colonna da Γ_{post} e Γ_{-i} indica la i-esima colonna di Γ_{post} senza l'elemento della i-esima riga.

Presentati tutti gli ingredienti necessari all'aggiornamento dei parametri e alla stima della media a posteriori, è doveroso soffermarsi sulla quantità $\Phi_{n+m}(\cdot)$.

3.3.2 Funzione di ripartizione multivariata normale

Nonostante la strategia presentata non richieda simulazioni dalla posteriori per ottenere una stima dei parametri, è d'obbligo capire se il calcolo di $\Phi_n(\cdot)$ possa considerarsi un ostacolo, dato che un suo valore esatto non è ottenibile per via analitica. In casi di piccoli valori di n si usano tecniche di $quasi - randomized$ Monte Carlo (Genz & Bretz, 2009) le quali permettono accurate valutazioni di $\Phi_n(\cdot)$. Quando n cresce, grazie al contributo di Botev (2017), si preferisce simulare via $minimax - tilting$ con il quale è possibile avere una stima efficiente di $\Phi_n(\cdot)$ con n di grandezza moderata. Il metodo può essere brevemente riassunto come un problema di simulazione esatta da una distribuzione troncata normale multivariata di dimensione n. Si basa essenzialmente sull' $exponential - tilting$, ossia una tecnica di traslazione della distribuzione, con lo scopo di risolvere un problema di ottimizzazione $minimax$ (punto di sella). L'ottimizzazione può essere risolta in modo efficiente perché sfrutta le proprietà di log-concavità della distribuzione normale.

In generale, sia Y una variabile casuale con densità h dipendente dal parametro θ,

la sua versione *exponentially tilded* $h_\theta(y)$ è uguale a

$$h_\theta(y) = \exp\{\theta y - K(\theta)\}h(y),$$

dove $K(\theta) = \log[\mathbb{E}[\{\exp(\theta Y)\}]$ è la funzione generatrice dei cumulanti di Y.

La tecnica dell'*exponentially tilded* unita a degli approcci di accetto-rifiuto migliora la proporzione di valori accettati dall'algoritmo rispetto agli usuali metodi Markov Chain Monte Carlo.

L'*exponentially tilded* viene particolarmente usata nella stima di eventi rari: nel caso in esame il valore della funzione $\Phi_n(\cdot)$ può assumere anche valori molto prossimi allo zero, rendendo una sua stima accurata non banale.

Per l'implementazione del metodo è stato utilizzato il pacchetto R `TruncatedNormal` (Botev & Belzile, 2021) il quale permette di assegnare un valore alla funzione di ripartizione della normale n-variata.

3.3.3 Problemi computazionali

Concentrandosi ora sulla (3.9), una volta che sopraggiungono nuovi dati è possibile ottenere una stima aggiornata del parametro β. Non banale è l'inizializzazione dei parametri della distribuzione a priori quando non si hanno partite a disposizione. Una possibilità è utilizzare ancora una volta una distribuzione normale poiché, oltre ad avere il vantaggio di non essere un forte vincolo per i parametri, è anche un caso speciale della SUN, rendendo dunque possibile l'aggiornamento dei parametri della priori come descritto nel Paragrafo 3.3.1. Ponendo $\Delta = 0_{p\times n}$, $\gamma = 0_n$ e $\Gamma = I_n$, sostituendo nella (3.8) ci si riconduce di fatto alla $N_p(\xi, \Omega)$. La posteriori appartiene dunque alla famiglia SUN (Durante, 2019), in particolare assumendo nuovamente per i dati il modello *probit* e la distribuzione a priori $\beta \sim N_p(\xi, \Omega)$ si ha che

$$\beta|X, y \sim SUN_{p,n}(\xi_{\text{post}}, \Omega_{\text{post}}, \Delta_{\text{post}}, \gamma_{\text{post}}, \Gamma_{\text{post}}),$$

con $\xi_{\text{post}} = \xi$, $\Omega_{\text{post}} = \Omega$, $\Delta_{\text{post}} = \bar{\Omega}\omega D^T s^{-1}$, $\gamma_{\text{post}} = s^{-1}D\xi$ e $\Gamma_{\text{post}} = s^{-1}(D\Omega D^T + I_n)s^{-1}$.

A questo punto si è proceduto in maniera analoga a quanto presentato nel Paragrafo 3.2.1, ossia la posteriori calcolata in una generica data verrà utilizzata come priori per la data successiva e, di volta in volta, verrà calcolata la media a posteriori della distribuzione. Qui sorgono due problemi non banali. Per il calcolo del valore atteso della distribuzione a posteriori in (3.9) particolarmente problematico risulta il parametro ψ; si supponga per semplicità di essere a ridosso della prima partita e di assumere $m = 0$. Il

calcolo di ψ richiede, per ogni $i = 1, ..., n$, con n numero di osservazioni disponibili per la prima giornata di gioco, il calcolo di $\Phi_{n-1}(\cdot)$. Di conseguenza ogni valore atteso necessita della valutazione di n funzioni di ripartizione della normale $(n-1)$-variata, oltre a una valutazione di $\Phi_n(\cdot)$. Nell'ambito del presente studio, dove le giornate variano tra i 500 e i 5000 eventi, il metodo risulta impraticabile, poiché con n dell'ordine delle centinaia una singola valutazione di $\Phi_n(\cdot)$ può richiedere diversi minuti. Il secondo ostacolo è ancora più vincolante, infatti la dimensione delle matrici Δ, Ω e del vettore γ cresce al crescere del numero di osservazioni e, di conseguenza, cresce la dimensione della normale multivariata della quale calcolare la funzione di ripartizione.

Esistono particolari situazioni dove il calcolo di $\Phi_n(\cdot)$ risulta semplificato, ad esempio utilizzando una matrice di varianza e covarianza diagonale. In casi del genere, il calcolo della funzione di ripartizione n-variata si riconduce al prodotto di n funzioni di ripartizioni univariate $\Phi(\cdot)$, rendendo la procedura molto più efficiente. Tuttavia, in questo caso la natura del metodo vieta questa semplificazione, dato che le matrici appena citate si popolano interamente man mano che arrivano nuovi dati.

Per ridurre lo sforzo computazionale si potrebbe pensare di ridurre la dimensione dello spazio parametrico escludendo gli effetti fissi per i giocatori, ma come è stato precedentemente chiarito il maggior problema si riscontra nella valutazione di $\Phi_n(\cdot)$ al crescere di n, dunque ridurre il numero di variabili non porterebbe a benefici.

Riprendendo i risultati di Durante (2019), esiste una forma esplicita anche per la distribuzione predittiva a posteriori. Supponendo di essere interessati a prevedere l'esito di una nuova osservazione y_{new} con relativo vettore di esplicative x_{new}, assumendo ancora una volta $m = 0$ e $\beta \sim N_p(\xi, \Omega)$ risulta

$$pr(y_{\text{new}} = 1|X, y, x_{\text{new}}) = \frac{\Phi_{n+1}\{s_{\text{new}}^{-1} D_{\text{new}} \xi; s_{\text{new}}^{-1}(D_{\text{new}} \Omega D_{\text{new}}^T + I_{n+1}) s_{\text{new}}^{-1}\}}{\Phi_n\{s^{-1} D\xi; s^{-1}(D\Omega D^T + I_n) s^{-1}\}},$$

dove D_{new} indica la matrice $(n+1) \times p$ ottenuta aggiungendo una riga $d_{\text{new}}^T = x_{\text{new}}^T$ a D e $s_{\text{new}} = \text{diag}\{(d_1^T \Omega d_1)^{0.5}, ..., (d_n^T \Omega d_n)^{0.5}, (d_{\text{new}}^T \Omega d_{\text{new}} + 1)^{0.5}\}$.

Purtroppo anche in questo caso, per ogni nuova osservazione è necessario il calcolo di due $\Phi_n(\cdot)$ il che risulta proibitivo al crescere di n, portando a scartare anche questa opzione.

Un'ulteriore strada prevede invece di generare valori direttamente dalla distribuzione a posteriori. Grazie ancora una volta ai contributi di Durante (2019) e sfruttando lo schema di campionamento *minimax tilting* e accetto-rifiuto proposto da Botev (2017) è possibile ottenere un campione di valori indipendenti dalla SUN a posteriori. Anche

questo metodo però alterna simulazioni da una normale p-variata e una normale troncata n-variata, non risolvendo i reali problemi sopracitati.

In conclusione i metodi presentati in questi paragrafi sono interessanti dal punto di vista teorico, ma di fatto impraticabili nel caso in esame. È necessario dunque cercare altre strade che riducano la complessità del metodo e guadagnino in termini di efficienza.

3.4 Algoritmo EP

Appurata l'inapplicabilità del metodo con priori coniugata per il modello *probit*, è necessario trovare un'altra strategia che consenta di trattare le distribuzioni a priori e a posteriori in maniera più agevole e che non si basi su risultati asintotici. Una possibilità è quella di utilizzare l'algoritmo EP (*expectation-propagation*), il quale verrà prima presentato nella sua versione generale e poi contestualizzato al caso di studio con regressione *probit*. L'algoritmo ha lo scopo di fornire un'approssimazione di distribuzioni non appartenenti a una famiglia nota oppure, come per la distribuzione a posteriori presentata nel Paragrafo 3.3.1, di difficile utilizzo.

Supponendo dunque di essere interessati a una generica distribuzione $f(\theta)$ fattorizzabile come

$$f(\theta) \propto \prod_{i=0}^{n} f_i(\theta),$$

l'algoritmo procede rimpiazzando iterativamente $f(\theta)$ con una sua approssimazione $g(\theta)$ la quale ammette la stessa fattorizzazione:

$$g(\theta) \propto \prod_{i=0}^{n} g_i(\theta).$$

Ad ogni iterazione dell'algoritmo e per ogni $i = 0, ..., n$ si prende l'approssimazione $g(\theta)$ e si sostituisce $g_i(\theta)$ con il corrispondente fattore $f_i(\theta)$ dalla distribuzione di interesse. Si definisce dunque la *cavity distribution* come

$$g_{-i}(\theta) \propto \frac{g(\theta)}{g_i(\theta)},$$

ossia la distribuzione approssimante $g(\theta)$ privata dell'i-esimo fattore, mentre la *tilted distribution*

$$g_{\setminus i}(\theta) \propto f_i(\theta) g_{-i}(\theta).$$

Algoritmo 2 Algoritmo EP

Input: $g_0(\theta), ..., g_n(\theta)$

while tutte le $g_i(\theta)$ convergono **do**
  for i in $0 : n$ **do**
    1: Calcolo della $g_{-i}(\theta) \propto g(\theta)/g_i(\theta)$
    2: Aggiornamento della $g_i(\theta)$ tale che $g_i(\theta)g_{-i}(\theta) \simeq f_i(\theta)g_{-i}(\theta)$
  end for
end while

Output: $g_0(\theta), ..., g_n(\theta)$ aggiornati

cioè la cosiddetta distribuzione ibrida, nella quale l'i-esimo fattore $g_i(\theta)$ è rimpiazzato dalla distribuzione obiettivo $f_i(\theta)$.

L'algoritmo procede costruendo prima un'approssimazione $g^*(\theta)$ della *tilded distribution* con la quale è poi possibile ottenere una approssimazione della distribuzione obiettivo $f_i(\theta)$ come $g_i^*(\theta) \propto g^*(\theta)/g_{-i}(\theta)$. Iterando questi due passaggi fino a convergenza si ottiene lo schema visibile nell' Algoritmo 2. Il secondo *step* dell'algoritmo può essere visto in maniera più compatta come la ricerca del $g_i(\theta)$ che renda la divergenza di Kullback-Leibler tra $f_i(\theta)g_{-i}(\theta)$ e $g_i(\theta)g_{-i}(\theta)$ minima (Vehtari et al., 2020).

Se si assume che $g(\theta)$ e $g_i(\theta)$, $i = 0, ..., n$ appartengano alla famiglia esponenziale, l'algoritmo risulta estremamente efficiente. Infatti, qualsiasi prodotto o divisione tra queste distribuzioni risulta ancora una famiglia esponenziale, ottenendo risultati per via analitica.

Riprendendo ancora il secondo passaggio dell'algoritmo, tipicamente si procedente con un *moment matching*, ossia con un confronto dei momenti tra la distribuzione $f_i(\theta)g_{-i}(\theta)$ e $g_i(\theta)g_{-i}(\theta)$ (Chopin & Ridgway, 2017). Questo corrisponde a minimizzare la divergenza di Kullback-Leibler tra $g_{\setminus i}(\theta)$ e $g(\theta)$ (Vehtari et al., 2020). L'attenzione si sposta di conseguenza verso quelle distribuzioni individuabili da un numero finito di momenti.

3.4.1 EP e distribuzione a posteriori

Tornando al caso in esame, il principale problema riscontrato nell'utilizzo della distribuzione coniugata per il modello *probit* risulta nella difficoltà di utilizzare agevolmente le distribuzioni esatte (Paragrafo 3.3.3). È naturale allora cercare di ricavare un'approssimazione per la distribuzione a posteriori tramite l'algoritmo EP. La funzione obiettivo risulta dunque la distribuzione a posteriori $\pi(\beta|X, y)$, la quale rispetta il criterio di fattorizzazione presentato nel Paragrafo 3.4, poiché

$$\pi(\beta|X,y) \propto \pi(\beta)\prod_{i=1}^{n} f(y_i|\beta),$$

ricordando che $f(y_i|\beta)$ è il contributo dell'osservazione i-esima alla verosimiglianza. Una scelta usuale per approssimare la distribuzione è utilizzare una $g(\beta) = \prod_{i=0}^{n} g_i(\beta)$ tale che $g_0 = \pi(\beta)$ riconducendosi a

$$g(\beta) \propto \pi(\beta)\prod_{i=1}^{n} g_i(\beta).$$

Si sceglie anche in questo caso una distribuzione normale per $g(\beta)$, sfruttando il vantaggio, tra le altre cose, di appartenere a una famiglia esponenziale e di essere completamente identificata dai primi due momenti. Si assume dunque che $g_i(\beta) \propto \exp\{-\frac{1}{2}\beta^T Q_i\beta + \beta^T r_i\}$, per $i = 1, ..., n$ e $g_0(\beta) \propto \exp\{-\frac{1}{2}\beta^T\beta\}$, ossia densità gaussiane scritte in forma esponenziale regolare. Di conseguenza è possibile osservare che $g(\beta)$ assume la forma di una $N_p(Q^{-1}r, Q^{-1})$, con $Q = \sum_{i=1}^{n} Q_i$ e $r = \sum_{i=1}^{n} r_i$. A questo punto, come descritto nel Paragrafo 3.4, sfruttando l'algoritmo EP si procede con un aggiornamento graduale dei momenti di ogni fattore $g_i(\beta)$ tramite un *moment matching* tra la distribuzione ibrida

$$g_{\setminus i}(\beta) = f(y_i|\beta)\prod_{j\neq i} g_j(\beta) \propto \Phi\{(2y_i - 1)\boldsymbol{x_i}^T\beta\}\prod_{j\neq i} g_j(\beta) \tag{3.10}$$

e l'approssimazione $g(\beta)$. Il calcolo dei primi due momenti della distribuzione $g_{\setminus i}(\beta)$ richiede lo svolgimento di integrali p-dimensionali, poiché

$$\begin{aligned}
\mu_{g_{\setminus i}} &= \mathbb{E}_{g_{\setminus i}(\beta)}[\beta] = \frac{1}{C_{g_{\setminus i}}}\int_{\mathbb{R}^p} \beta f(y_i|\beta)\prod_{j\neq i} g_j(\beta)\ d\beta,\\
\Sigma_{g_{\setminus i}} &= \mathbb{V}_{g_{\setminus i}(\beta)}[\beta] = \frac{1}{C_{g_{\setminus i}}}\int_{\mathbb{R}^p} \beta\beta^T f(y_i|\beta)\prod_{j\neq i} g_j(\beta)\ d\beta, \quad \textit{con}\\
C_{g_{\setminus i}} &= \int_{\mathbb{R}^p} f(y_i|\beta)\prod_{j\neq i} g_j(\beta)\ d\beta,
\end{aligned}$$

dove $C_{g_{\setminus i}}$ è la costante di normalizzazione. Una volta ottenuti i momenti della distribuzione $g_{\setminus i}(\beta)$ vengono aggiornati i parametri del fattore $g_i(\beta)$ escluso dal calcolo, come:

$$Q_i = \Sigma_{g_{\setminus i}}^{-1} - Q_{-i}, \quad r_i = \Sigma_{g_{\setminus i}}^{-1}\mu_{g_{\setminus i}} - r_{-i}, \tag{3.11}$$

escludendo di volta in volta un $g_i(\beta)$ diverso e ripetendo il *moment matching* fino a convergenza. È importante sottolineare che il primo fattore $g_0(\beta)$, ossia quello posto uguale alla distribuzione a priori $\pi(\beta)$, non verrà mai aggiornato durante l'intero algoritmo, permettendo così all'approssimazione a posteriori finale di essere in parte influenzata dall'informazione a priori.

Il metodo richiede dunque il calcolo di integrali non sempre risolvibili per via analitica, costringendo all'utilizzo di metodi numerici come, ad esempio, quadrature gaussiane, rendono i calcoli onerosi specialmente al crescere di p. Tuttavia, grazie ai recenti contributi di Fasano et al. (2023), nella (3.10) è possibile riconoscere il nucleo di una distribuzione normale asimmetrica estesa $SN_p(\xi_i, \Omega_i, \alpha_i, \tau_i)$ (Azzalini & Capitanio, 2014), un ampliamento, tramite l'aggiunta di un parametro, della distribuzione SN definita nel Paragrafo 3.3. Nel caso in esame si ha in particolare che

$$\begin{aligned} \xi_i &= Q_{-i}^{-1} r_{-i}, & \Omega_i &= Q_{-i}^{-1}, \\ \alpha_i &= (2y_i - 1)\omega_i \boldsymbol{x_i}, & \tau_i &= (2y_i - 1)(1 + \boldsymbol{x_i}^T \Omega_i \boldsymbol{x_i})^{-0.5} \boldsymbol{x_i}^T \xi_i, \end{aligned}$$

con $Q_{-i} = \sum_{j \neq i} Q_j$, $r_j = \sum_{j \neq i} r_j$ e $\omega_i = (\text{diag}(\Omega_i))^{0.5}$. Il risultato è estremamente vantaggioso, poiché non richiede l'utilizzo di metodi numerici per il calcolo degli integrali definiti precedentemente, essendo i momenti della normale asimmetrica disponibili in forma chiusa:

$$\begin{aligned} \mu_{g_{\setminus i}} &= \mathbb{E}_{g_{\setminus i}(\beta)}[\beta] = \xi_i + \zeta_1(\tau_i) s_i \Omega_i \boldsymbol{x_i}, \\ \Sigma_{g_{\setminus i}} &= \mathbb{V}_{g_{\setminus i}(\beta)}[\beta] = \Omega_i + \zeta_2(\tau_i) s_i^2 (\Omega_i \boldsymbol{x_i})(\Omega_i \boldsymbol{x_i})^T, \end{aligned}$$

dove $s_i = (2y_i - 1)(1 + \boldsymbol{x_i}^T \Omega_i \boldsymbol{x_i})^{-0.5}$. Richiamando ancora una volta i risultati di Azzalini & Capitanio (2014), le funzioni $\zeta_1(\cdot)$ e $\zeta_2(\cdot)$ possono essere espresse come

$$\zeta_1(x) = \phi(x)/\Phi(x), \quad \zeta_2(x) = -\zeta_1(x)^2 - x\zeta_1(x).$$

A questo punto è possibile proseguire con un *moment matching* tra la distribuzione ibrida e l'approssimazione normale senza ricorre a calcoli numerici. Siano dunque Q_i^{new} e r_i^{new} i parametri del generico fattore $g_i(\beta)$ aggiornati dopo il *moment matching*. Per quanto appena discusso, devono valere le uguaglianze

$$\begin{cases} (Q_{-i} + Q_i^{\text{new}})^{-1} = \Sigma_{g_{\setminus i}}, \\ (Q_{-i} + Q_i^{\text{new}})^{-1}(r_{-i} + r_i^{\text{new}}) = \mu_{g_{\setminus i}}, \end{cases}$$

e isolando le quantità di interesse

$$\begin{cases} Q_i^{\text{new}} = \Sigma_{g_{\setminus i}}^{-1} - Q_{-i}, \\ r_i^{\text{new}} = \Sigma_{g_{\setminus i}}^{-1}\mu_{g_{\setminus i}} - r_{-i}, \end{cases}$$

riconducendosi di fatto allo stesso sistema visibile nella (3.11). Eguagliare i momenti delle due distribuzioni significa permettere ai parametri Q_i^{new} e r_i^{new} di muoversi nella direzione indicata dai dati, modificando gradualmente l'intera distribuzione $g(\beta)$.

Per evitare il calcolo diretto dell'inversa della matrice $\Sigma_{g_{\setminus i}}^{-1}$ di dimensioni $p \times p$ è possibile sfruttare l'identità di Woodbury:

$$\begin{aligned} Q_i^{\text{new}} &= (\Omega_i + \zeta_2(\tau_i)s_i^2(\Omega_i\boldsymbol{x_i})(\Omega_i\boldsymbol{x_i})^T)^{-1} - Q_{-i} \\ &= \Omega_i^{-1} - \zeta_2(\tau_i)\, s_i^2 \left(1 + \zeta_2(\tau_i)\, s_i^2 \boldsymbol{x}_i^\top \Omega_i \Omega_i^{-1} \Omega_i \boldsymbol{x}_i\right)^{-1} \Omega_i^{-1}\Omega_i \boldsymbol{x}_i \boldsymbol{x}_i^\top \Omega_i \Omega_i^{-1} - Q_{-i} \\ &= -\zeta_2(\tau_i)\, s_i^2 \left(1 + \zeta_2(\tau_i)\, s_i^2 \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i\right)^{-1} \boldsymbol{x}_i \boldsymbol{x}_i^\top = -\left(\zeta_2(\tau_i)^{-1}\, s_i^{-2} + \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i\right)^{-1} \boldsymbol{x}_i \boldsymbol{x}_i^\top \\ &= -\frac{\zeta_2(\tau_i)}{1 + \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i + \zeta_2(\tau_i)\, \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i} \boldsymbol{x}_i \boldsymbol{x}_i^\top = k_i^{\text{new}} \boldsymbol{x}_i \boldsymbol{x}_i^\top, \end{aligned}$$

con $k_i^{\text{new}} = -\zeta_2(\tau_i) \,/\, \left(1 + \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i + \zeta_2(\tau_i)\, \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i\right)$. Sfruttando questo risultato si ha inoltre

$$\begin{aligned} \boldsymbol{r}_i^{\text{new}} &= Q_{-i}\mu_{g_{\setminus i}} + Q_i^{\text{new}}\mu_{g_{\setminus i}} - \boldsymbol{r}_{-i} = Q_{-i}Q_{-i}^{-1}\boldsymbol{r}_{-i} + \zeta_1(\tau_i)\, s_i Q_{-i}\Omega_i \boldsymbol{x}_i + Q_i^{\text{new}}\mu_{g_{\setminus i}} - \boldsymbol{r}_{-i} \\ &= \zeta_1(\tau_i)\, s_i \boldsymbol{x}_i + Q_i^{\text{new}}\mu_{g_{\setminus i}} = \zeta_1(\tau_i)\, s_i \boldsymbol{x}_i + k_i^{\text{new}} \boldsymbol{x}_i \boldsymbol{x}_i^\top \Omega_i \boldsymbol{r}_{-i} + k_i^{\text{new}} \zeta_1(\tau_i)\, s_i \boldsymbol{x}_i \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i \\ &= \left[\zeta_1(\tau_i)\, s_i + k_i^{\text{new}} \left(\Omega_i \boldsymbol{x}_i\right)^\top \boldsymbol{r}_{-i} + k_i^{\text{new}} \zeta_1(\tau_i)\, s_i \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i\right] \boldsymbol{x}_i = m_i^{\text{new}} \boldsymbol{x}_i, \end{aligned}$$

con $m_i^{\text{new}} = \zeta_1(\tau_i)\, s_i + k_i^{\text{new}} \left(\Omega_i \boldsymbol{x}_i\right)^\top \boldsymbol{r}_{-i} + k_i^{\text{new}} \zeta_1(\tau_i)\, s_i \boldsymbol{x}_i^\top \Omega_i \boldsymbol{x}_i$.

Dunque l'interesse si sposta verso il calcolo e la memorizzazione dei parametri k_i e m_i, necessari per ottenere le quantità aggiornate Q_i^{new} e r_i^{new}.

Per l'inizializzazione dei due parametri, una possibilità è quella di porli uguali a zero, in modo tale da utilizzare come prima approssimazione $g(\beta)$ la distribuzione a priori, per poi aggiornarli gradualmente. Nell'Algoritmo 3 è possibile osservare la sequenza di operazioni da svolgere: dopo aver calcolato le quantità Q_{-i}, r_{-i} e Ω_i escludendo

Algoritmo 3 EP per distribuzione a posteriori modello *probit*

Input: Q^{-1}, r, X, y

$k_1 = m_1 = 0_p$
while k e m convergono **do**
  for i in 1:n **do**
    $Q_{-i} = Q - k_i \boldsymbol{x_i} \boldsymbol{x_i}^T$ ▷ Vecchio valore di k_i
    $r_{-i} = r - m_i \boldsymbol{x_i}$ ▷ Vecchio valore di r_i
    $\Omega_i = Q^{-1} + k_i/(1 - k_i \boldsymbol{x_i}^T Q^{-1} \boldsymbol{x_i})(Q^{-1}\boldsymbol{x_i})(Q^{-1}\boldsymbol{x_i})^T$
    $s_i = (2y_i - 1)(1 + \boldsymbol{x_i}^T \Omega_i \boldsymbol{x_i})^{-0.5}$
    $\tau_i = s_i \boldsymbol{x_i}^T \Omega_i r_{-i}$
    $k_i = -\zeta_2(\tau_i)/(1 + \boldsymbol{x_i}^T \Omega_i \boldsymbol{x_i} + \zeta_2(\tau_i)\boldsymbol{x_i}^T \Omega_i \boldsymbol{x_i})$ ▷ k_i aggiornato
    $m_i = \zeta_1(\tau_i) s_i + k_i(\Omega_i \boldsymbol{x_i})^T r_{-i} + k_i \zeta_1(\tau_i) s_i \boldsymbol{x_i}^T \Omega_i \boldsymbol{x_i}$ ▷ m_i aggiornato
    $Q = Q_{-i} + k_i \boldsymbol{x_i} \boldsymbol{x_i}^T$ ▷ *Moment matching*
    $r = r_{-i} + m_i \boldsymbol{x_i}$ ▷ *Moment matching*
    $Q^{-1} = \Omega_i + \zeta_2(\tau_i) s_i^2 (\Omega_i \boldsymbol{x_i})(\Omega_i \boldsymbol{x_i})^T$
  end for
end while

Output: Q^{-1}, r aggiornati

dunque il contributo dell'i-esimo fattore $q_i(\beta)$, si procede con un *update* dei parametri k_i e m_i, i quali verranno utilizzati per ottenere i valori di Q e r aggiornati tramite il *moment matching* descritto precedentemente. L'i-esimo valore di k e l'i-esimo valore di r verranno utilizzati nel successivo ciclo *for* come "vecchi" valori per ottenere nuovamente le quantità prive del contributo del fattore $q_i(\beta)$. Quando la procedura non porta più a un aggiornamento significativo di k e m l'algoritmo si interrompe restituendo il nuovo vettore di medie r e la nuova matrice di varianze e covarianze Q^{-1}.

Si noti come per il calcolo di Ω_i sia stata utilizzata ancora una volta l'identità di Woodbury, evitando così di dover invertire direttamente matrici dell'ordine $p \times p$:

$$\Omega_i = Q_{-i}^{-1} = (Q - k_i \boldsymbol{x_i} \boldsymbol{x_i}^T)^{-1} = Q^{-1} + \frac{k_i}{1 - k_i \boldsymbol{x_i}^T Q^{-1} \boldsymbol{x_i}} (Q^{-1}\boldsymbol{x_i})(Q^{-1}\boldsymbol{x_i})^T,$$

considerando inoltre che Q^{-1} è nota fin dall'inizio, essendo questa la matrice di varianze e covarianze della distribuzione a priori.

A questo punto, tornando al caso di *data stream* in esame, verrà utilizzato uno schema simile a quello presentato nel Paragrafo 3.2.1. Una volta conclusa una generica giornata di gioco t e ottenuti i parametri a posteriori r e Q^{-1}, questi verranno utilizzati per inizializzare la distribuzione normale a priori della giornata successiva. L'operazione è ammissibile dato che, come illustrato precedentemente, il fattore $q_0(\beta)$ non viene mai

aggiornato durante tutto il procedimento, permettendo così alla distribuzione finale di essere in parte influenzata dall'informazione a priori.

Come nota conclusiva, è importante sottolineare che l'algoritmo presentato è utilizzabile in casi dove $p < n$. Le ultime giornate dei Playoff non soddisfano questo criterio, essendo composte da meno di 470 eventi. Per questo motivo si è deciso di accorpare le giornate con numero di eventi insufficienti, passando da 213 giornate a 201. Si noti che una versione dell'algoritmo in casi di $p > n$ è disponibile in Fasano et al. (2023). Tuttavia, si basa sull'assunzione che la matrice di varianze e covarianze della distribuzione a priori in *input* sia diagonale, ipotesi troppo stringente per il caso in esame, poiché la matrice Q^{-1} si popola gradualmente con l'avanzare degli incontri.

3.4.2 Classifica EP e normale-normale

Discusso l'algoritmo è possibile presentare la classifica finale che ne deriva. Si seguirà ancora una volta la stessa procedura adottata per gli altri metodi, ossia calcolo delle stime di pericolosità degli eventi, applicazione della media per giocatore, aggiustamento per gli eventi al minuto, lisciamento tramite media mobile e applicazione della mediana, come discusso nel Paragrafo 3.2.3 e nel Capitolo 2. In Figura 3.4 vengono riportati ancora una volta gli andamenti di 4 coefficienti all'avanzare del campionato dati dal modello con l'utilizzo dell'algoritmo EP e dal modello con approssimazione normale senza l'utilizzo della *power prior* (Paragrafo 3.2.1). Quello che salta subito all'occhio è che, a parte per le prime giornate dove si notano delle leggere differenze, i due metodi sembrerebbero fornire valori molto simili. Questo non sorprende poiché per entrambi gli approcci vengono aggiornati i parametri di una distribuzione normale utilizzata come priori. Il modello normale-normale, tuttavia, si basa sull'approssimazione asintotica: per le prime giornate di campionato l'assunzione potrebbe non essere del tutto corretta, giustificando le leggere differenze con il modello EP, che, d'altra parte, si basa su un confronto tra momenti di due distribuzioni, non servendosi dunque di risultati asintotici.

Nel concreto, le differenze tra i due approcci sono minime e i risultati in Tabella 3.3 lo confermano. Ad eccezione di alcuni valori leggermente diversi del punteggio finale, i due metodi sembrerebbero fornire gli stessi migliori 15 *top player* della stagione. Rispetto alla classifica ottenuta con il modello con correzione di Firth (Paragrafo 2.3) si notano, tuttavia, delle differenze. Giannis Antetokounmpo sale in vetta alla classifica mentre guadagnano alcune posizioni Kevin Durant, Kawhi Leonard e Lebron James, piazzandosi tra la quinta e la decima posizione. Scende di una posizione Jonas Valanciunas mentre esce dalla *top* 10 Jusuf Nurkic.

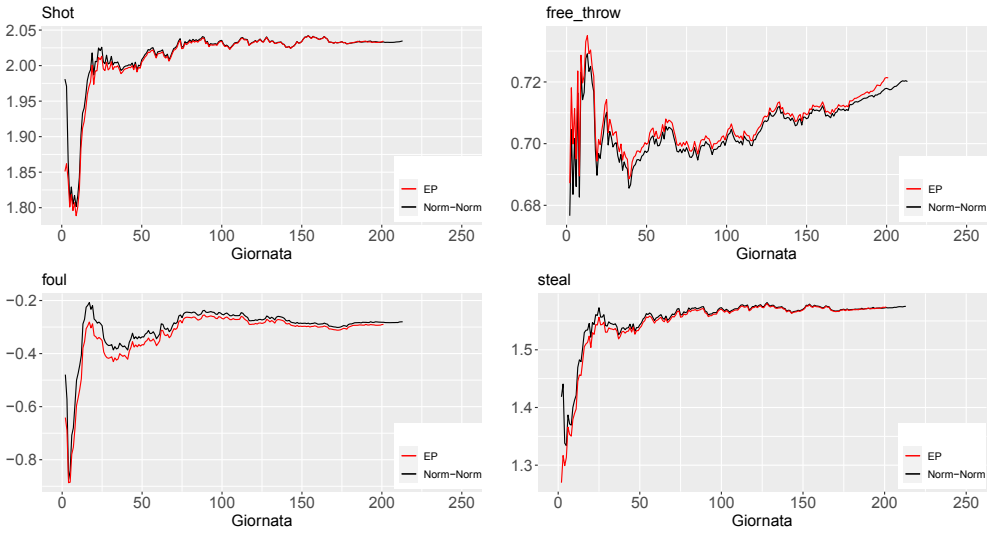

FIGURA 3.4: Andamento dei parametri `Shot` `free_throw`, `foul` e `steal` per il modello con aggiornamento normale-normale e il modello con algoritmo EP

normale-normale		**EP**	
Player	**Score**	**Player**	**Score**
Giannis Antetokounmpo	0.8494	Giannis Antetokounmpo	0.8486
Joel Embiid	0.8401	Joel Embiid	0.8401
Anthony Davis	0.7739	Anthony Davis	0.7739
James Harden	0.7483	James Harden	0.7483
Kawhi Leonard	0.7180	Kawhi Leonard	0.7175
Jonas Valanciunas	0.6801	Jonas Valanciunas	0.6800
Kevin Durant	0.6779	Kevin Durant	0.6777
Nikola Jokic	0.6759	Nikola Jokic	0.6758
LeBron James	0.6735	LeBron James	0.6734
Julius Randle	0.6624	Julius Randle	0.6623
Andre Drummond	0.6585	Andre Drummond	0.6585
Russell Westbrook	0.6573	Russell Westbrook	0.6573
Jusuf Nurkic	0.6539	Jusuf Nurkic	0.6532
Karl-Anthony Towns	0.6488	Karl-Anthony Towns	0.6485
Paul George	0.6434	Paul George	0.6434

TABELLA 3.3: Classifica migliori 15 giocatori con modello normale-normale e modello EP a fine campionato

Nel complesso, ad eccezione delle prime 4 posizioni, la nuova classifica sembrerebbe selezionare giocatori più abili rispetto a quella ottenuta con il modello di Firth, ricordando che proprio nella stagione 2018-19 Kawhi Leonard vince il titolo di MVP delle finali, mentre Kevin Durant e Lebron James sono considerati due tra i giocatori più forti della storia del *basket*.

Si noti infine come l'aggiunta degli effetti fissi per giocatore nel modello normale-normale (Tabella 3.3) porti a un miglioramento, in termini di giocatori selezionati, rispetto alla classifica dello stesso modello considerando gli effetti fissi per squadra (Tabella 3.2), giustificando, ancora una volta, l'inserimento di questi parametri.

3.5 Valutazioni

Le metodologie bayesiane discusse in questo capitolo si sono rivelate adatte per l'obiettivo della tesi. Sono stati affrontati con efficacia i problemi legati alla dimensione dello spazio campionario e dello spazio parametrico. Gli approcci tramite utilizzo dell'approssimazione normale e dell'algoritmo EP hanno fornito essenzialmente gli stessi risultati. Questo non sorprende dato che entrambi aggiornano i parametri di una distribuzione normale, con metodi tuttavia differenti. L'inserimento della *power prior* si è rilevato una alternativa interessante, tuttavia a causa dell'elevato carico computazionale si è deciso di non inserire gli effetti fissi per giocatore tra le variabili del modello. L'utilizzo delle verosimiglianze pesate non ha portato a uno stravolgimento della classifica rispetto al modello privo di pesi, fornendo un interessante punto di partenza per riflessioni future.

Conclusioni

Con il presente elaborato si è voluto esplorare il complesso mondo del *data stream* in un contesto sportivo, come quello del campionato NBA nella stagione 2018/19. Il principale scopo della tesi è stato quello di aggiornare dinamicamente la classifica dei migliori giocatori della stagione attraverso un indice di pericolosità offensiva creato a partire dalle azioni di gioco. Sono stati presentati e approfonditi diversi approcci, sia in termini di complessità, sia in termini di metodo.

In un primo approccio è stato utilizzato un modello di regressione binaria con *link probit* e correzione di Firth. Ad ogni nuova giornata di gioco, è stato adattato un nuovo modello sui soli dati disponibili dalla giornata corrente e ottenute le stime di pericolosità per ogni evento.

In un secondo approccio, sono state utilizzate delle tecniche bayesiane, con lo scopo di ottenere delle stime dei parametri che, a differenza del metodo con correzione di Firth, fossero in parte influenzate dalle partite precedenti, oltre che dai dati disponibili per la giornata di interesse. Questo è stato possibile utilizzando la distribuzione a posteriori di una generica giornata come distribuzione a priori per la giornata successiva. Inoltre, per ridurre il peso delle partite distanti, in termini di giornate di gioco, rispetto a quella di interesse, sono stati inseriti dei pesi per le verosimiglianze proporzionali alla distanza temporale tra le partite, tramite la tecnica della *power prior*.

Alle stime di probabilità degli eventi, per tutti gli approcci seguiti, sono state successivamente applicate delle medie per giocatore, di giornata in giornata, in modo da ottenere delle prime stime di pericolosità offensiva degli atleti. Per aggregare i risultati e includere della dipendenza temporale legata allo stato di forma dei giocatori, i punteggi sono stati aggiustati per il numero di eventi al minuto effettuato da ogni giocatore fino a quel momento e lisciati tramite una media mobile pesata.

Grazie alla presenza di informazioni aggiuntive, come il ruolo dei giocatori, la squadra di appartenenza e il periodo della stagione, è stato possibile eseguire delle analisi stratificate. È emerso che i metodi utilizzati non sembrano essere adatti a fornire un indice di pericolosità offensiva per squadra. D'altra parte, una stratificazione per ruolo e

per periodo di gioco sembrerebbe necessaria, poiché alcuni ruoli hanno punteggi più alti di altri e tra la Regular Season e i Playoff le prestazioni dei giocatori possono variare.

L'approccio con il modello di Firth utilizzato ha il vantaggio di essere di facile implementazione, tuttavia risulta semplicistico per l'obiettivo della presente tesi, poiché non considera alcun tipo di dipendenza temporale tra le partite. Nonostante ciò, il modello riporta dei buoni risultati.

Gli approcci bayesiani, d'altra parte, si sono rivelati adatti per lo studio in esame. In particolare, sia con l'utilizzo del modello con distribuzione a priori e a posteriori normale, sia con l'utilizzo dell'algoritmo EP, è stato possibile fronteggiare con successo il problema legato alla dimensione dello spazio parametrico e dello spazio campionario. I risultati ottenuti, inoltre, possono essere ritenuti soddisfacenti.

L'inserimento della *power prior*, d'altra parte, ha portato a un inevitabile rallentamento della procedura di stima, rendendo necessaria la rimozione degli effetti fissi per giocatore. Discorso analogo vale per l'utilizzo della distribuzione SUN, coniugata del modello *probit*, la quale ha reso impraticabile ogni tipo di procedura.

Possibili sviluppi futuri potrebbero riguardare la ricerca di un modo per inserire dell'asimmetria nella distribuzione a posteriori dei parametri, sfruttando le relazioni tra il modello *probit* e la distribuzione SUN, tramite l'utilizzo di approssimazioni più sofisticate della normale.

Un altro possibile approfondimento riguarda la natura del modello con l'utilizzo della *power prior*: dal presente studio, infatti, non è del tutto chiaro come mai le stime di pericolosità offensiva dei giocatori, una volta applicata la media, siano così simili al modello privo dell'utilizzo dei pesi, nonostante i valori dei coefficienti siano molto diversi tra i due approcci. Sarebbe inoltre interessante capire se queste differenze rimangono inserendo gli effetti fissi per giocatore anche nel modello con *power prior*.

Appendice

Codice R utilizzato

Capitolo 1

Per questo capitolo viene riportata solamente la funzione per generare la variabile risposta definita nel Paragrafo 1.3.2

```
#klag: numero di lag da considerare
risposta=function(klag, dati){
  y=NULL
  fight=unique(dati$game_id)
  for(i in 1:length(fight)){
    if(i%%10==0){print(i)}
    small=dati[dati$game_id==fight[i],]
    period=unique(small$period)
    for(j in period){
      k=klag
      small2=small[small$period==j,]
      for(w in 1:nrow(small2)){
        flag=F
        if(w+k>nrow(small2)){
          k=k-1
        }
        for(h in 0:k){
          if(small2$result[w+h]=="made" &
             small2$team[w+h]==small2$team[w]){
            y=c(y,1)
            flag=T
            break
          }
          if(small2$team[w+h]!=small2$team[w]){
            y=c(y,0)
```

```
            flag=T
            break
          }
        }
        if(!flag){
          y=c(y,0)
        }
      }
    }
  }
  return(y)
}
```

Capitolo 2

Si riporta di seguito la funzione creata per calcolare di giornata in giornata il modello di regressione binaria con correzione di Firth (Paragrafo 2.2), fornire le previsioni e calcolare la media per giocatore

```
firth=function(Mdat, nuove){
  giorni=unique(Mdat$date)
  mmat=data.frame(player=unique(Mdat$player))
  for(i in 1:length(giorni)){
    dati=Mdat[Mdat$date==giorni[i],]
    rari=names(table(dati$player)>5)
    dati=dati[dati$player%in%rari, ]
    player=dati$player
    dati=dati[,c(32,45,nuove,95)]
    ap=apply(dati,2,unique)
    cc=NULL

    for(j in 1:length(ap)){
      if(length(ap[[j]])==1){
        cc=c(cc, names(ap[j]))
      }
    }
    via=which(names(dati)%in%cc)
    print(via)
    if(length(via)==0){
      mod=brglm(y3~., data=dati, family="binomial"(link = "probit"))
      pred=predict(mod, type="response")
```

```
      medie=tapply(pred, player, mean)
    }

    else{
      mod=brglm(y3~., data=dati[,-via], family="binomial"(link = "
   probit"))
      pred=predict(mod, type="response")
      medie=tapply(pred, player, mean)
    }
    medie=as.data.frame(medie)
    medie$player=rownames(medie)
    colnames(medie)=c(paste0("t",i),"player")
    rownames(medie)=NULL
    mmat=left_join(mat, medie, by="player")
    cat(i/length(giorni)*100, "\n")
  }
  return(mmat)
}
```

Di seguito vengono riportate le funzioni che calcolano il coefficiente aggiustato per ogni giocatore e lo moltiplicano per il *ranking* dato dal modello di Firth (Paragrafo 2.2.2) e la funzione che aggiusta per la media mobile pesata (Paragrafo 2.2.3)

```
#Calcolo dei coefficienti
adjScore=function(Mmat, Mdat){
  l=list()
  quando=unique(Mdat$date)
  for(i in 1:nrow(Mmat)){
    totA=totS=NULL
    pl=Mmat$player[i]
    fight=which(!is.na(Mmat[i,-1]))
    for(j in fight){
      dati=Mdat[Mdat$date==quando[j],] #dataset con una data dove
                                        il giocatore i ha giocato
      v=NULL
      for(k in 1:nrow(dati)){
        v=c(v, pl%in%dati[k,4:13]) #giocatori a1-a5 h1-h5
      }
      totA=c(totA,sum(dati$player==pl))
      totS=c(totS,sum(dati[v,"play_length"])/60)
    }
    cat(i/nrow(Mmat)*100,"\n")
    coef=cumsum(totA)/cumsum(totS)
```

```
    l$n=coef
    names(l)[i]=pl
  }
  return(l)
}


#Prodotto tra i vecchi punteggi e i coefficienti
newscore=function(Mmat, coef){
  vet=NULL
  m=matrix(NA, nrow(Mmat), ncol(Mmat))
  colnames(m)=colnames(Mmat)
  for(i in 1:nrow(Mmat)){
    dove=which(!is.na(Mmat[i,-1]))
    vet=as.numeric(Mmat[i,-1][dove]*coef[[i]])
    m[i,dove+1]=vet
  }
  m[,1]=Mmat$player
  return(as.data.frame(m))
}


#Lisciamento a media mobile
mm=function(score,pesi,lag){
  l=list()
  for(i in 1:nrow(score)){
    vet=as.numeric(score[i,-1][!is.na(score[i,-1])])
    for(j in 1:length(vet)){
      if(j>=lag){
        vet[j] <- sum(vet[(j - lag + 1):j] * pesi)
      }
    }
    l$pl=vet
    names(l)[i]=score[i,1]
  }
  return(l)
}



#Risultati finali
final=function(l,f){ #f funzione per aggregare i risultati
  su=NULL
  quanti=NULL
```

```
  for(i in 1:length(l)){
    su=c(su, f(l[[i]]))
    quanti=c(quanti, length(l[[i]]))
  }
  M=data.frame(player=names(l), sum=su, quanti=quanti)
  M=M[order(M$sum, decreasing = T),]
  M
}



#Punteggio per squadra
squadre=function(quali,a,Mdat){
  quando=unique(dati$date)
  b=NULL
  l=list()
  z=0
  for(i in quali){ #solo per i giocatori con piu' di una squadra
    b=NULL
    z=z+1
    chi=names(a[i])
    print(chi)
    dove=which(matdef$player==chi)
    fight=which(!is.na(matdef[dove,-1]))
    for(j in fight){
      dat=Mdat[Mdat$date==quando[j],]
      b=c(b,dat$team[dat$player%in%chi][1])
    }
    l$bo=b
    names(l)[z]=chi
    cat(z/length(quali)*100, "\n")
  }
  return(l)
}
```

Capitolo 3

Si riportano di seguito le funzioni create per il calcolo di tutte le quantità descritte nel Paragrafo 3.2, ossia la funzione per il calcolo della funzione punteggio e della matrice di informazione osservata per il modello *probit*, le stesse quantità aggiornate dopo l'arrivo

dei dati, l'algoritmo di Newton-Raphson e la funzione per la creazione della matrice di vettori β descritta nel Paragrafo 3.2.1

```
#U e J verosimiglianza
UJ_logv=function(beta, X, y){
  Z=Z2=matrix(0, length(beta),length(beta))
  V=V2=c(rep(0, length(beta)))
  for(i in 1:nrow(X)){
    etai=c(crossprod(X[i,],beta))
    xi=as.vector(X[i,])
    XX=tcrossprod(as.vector(X[i,]))
    etai=c(crossprod(X[i,],beta))
    phi=dnorm(etai)
    Phi=pnorm(etai)

    V=V+c(y[i]*Phi^(-1)*phi)*xi            #U
    V2=V2+(y[i]-1)*c((1-Phi)^(-1)*phi)*xi  #U
    Z=Z+y[i]*c((phi/Phi)^2+etai*phi/Phi)*XX    #J
    Z2=Z2+(y[i]-1)*(phi^2/(1-Phi)^2-etai*phi/(1-Phi))*XX    #J
  }
  return(list(U_logv=V+V2, J_logv=Z-Z2))
}


#U e J a posteriori
UJ_post=function(beta, beta0, sigma0, X, y){
  UJ=UJ_logv(beta, X, y)
  U=UJ$U_logv
  J=UJ$J_logv
  U_post=solve(sigma0)%*%(beta0-beta)+U
  J_post=solve(sigma0)+J
  return(list(U_post=U_post, J_post=J_post))
}


 #newton-raphson per la moda a posteriori
beta_post=function(eps, beta0, sigma0, X, y){
  diff=10
  bn=beta0
  while(any(diff>eps)){
    bn_1=bn
    UJ=UJ_post(bn_1, beta0, sigma0, X, y)
    U=UJ$U_post
```

```
    J=UJ$J_post
    bn=bn_1+solve(J)%*%U
    diff=abs(bn-bn_1)
  }
  bn
}

#aggiornamento bayesiano delle stime, restituisce i beta di giornata
   in giornata
aggbaye=function(beta0, sigma0, X, dat, eps){
y=dat$y3
  bete=NULL
  fight=unique(dat$date)
  for(i in 1:length(fight)){
    mat=X[which(dat$date==fight[i]),]
    risp=y[which(dat$date==fight[i])]
    betatilde=beta_post(eps, beta0, sigma0, mat, risp)
    beta0=betatilde #nuova priori per la giornata successiva
    sigma0=solve(UJ_post(betatilde,beta0, sigma0, mat, risp)$J_post)
    bete=cbind(bete, betatilde)
    colnames(bete)[i]=paste0("day_",i)
    cat(colnames(bete)[i], "\n")
  }
  return(bete)
}
```

Viene riportata di seguito la funzione per il calcolo della funzione punteggio e della matrice di informazione osservata a posteriori con la modifica tramite *power prior* descritta nel Paragrafo 3.2.2. Le altre funzioni rimangono invariate

```
alpha=function(x,k){
  exp(-k*x)
}

#i: giornata corrente
#k: parametro della funzione alpha()
#r: numero di ritardi massimo da considerare

UJ_postpp=function(beta, beta0, sigma0, X, y, dat, fight, i, k, r){
  w=max((i-r), 1)
  if(w==1 & i>1){
    a=c(alpha((i-1):1, k) ,1)
  }
```

```
  else if (w!=1 & i>1){
    a=c(alpha(r:1, k) ,1)
  }
  else{
    a=1
  }

  U=matrix(0, length(beta), 1)
  J=matrix(0, length(beta), length(beta))

  for(j in w:i){
    quali=which(dat$date==fight[j])
    ics=X[quali,]
    ipsilon=y[quali]
    UJ=UJ_logv(beta, ics, ipsilon)
    U=U+a[j-w+1]*UJ$U_logv
    J=J+a[j-w+1]*UJ$J_logv
    }
  Up=solve(sigma0)%*%(beta0-beta)+U
  Jp=solve(sigma0)+J
  return(list(U_post=Up, J_post=Jp))
}
```

Funzione che calcola le probabilità di ogni giornata e media per giocatore

```
prob=function(bete, mat, d){
  fight=unique(d$date)
  pi=NULL
  ret_m=data.frame(player=unique(d$player))
  for(i in 1:length(fight)){
    dat=mat[which(d$date==fight[i]),]
    d2=d[which(d$date==fight[i]),]
    pi=pnorm(dat%*%bete[,i])
    medie=tapply(pi, d2$player, mean)
    medie=as.data.frame(medie)
    medie$player=rownames(medie)
    colnames(medie)=c(paste0("t",i),"player")
    rownames(medie)=NULL
    ret_m=left_join(return_m, medie, by="player")
    cat(i/length(fight)*100, "\n")
  }
  return(ret_m)
}
```

Si riporta la funzione utilizzata per il calcolo dell'aggiornamento dei parametri della distribuzione SUN con distribuzione a priori Normale (Paragrafo 3.3.3) e la funzione per il calcolo della media a posteriori. La versione originale dei codici è disponibile al *link*: `https://github.com/danieledurante/ProbitSUN`

```
primoAGG=function(csi, omega, y, X){
  n=length(y)
  p=nrow(omega)
  D=diag(2*y-1)%*%X
  s=matrix(0,n,n)
  for(i in 1:n){
    di=D[i,]
    s[i,i]=sqrt(t(di)%*%omega%*%di+1)
  }
  om=diag(sqrt(diag(omega)))
  baromega=solve(om)%*%omega%*%solve(om)
  csiP=csi
  omegaP=omega
  deltaP=baromega%*%om%*%t(D)%*%solve(s)
  gammaP=solve(s)%*%D%*%csi
  GammaP=solve(s)%*%(D%*%omega%*%t(D)+diag(n))%*%solve(s)
  return(list(csiPost=csiP, omegaPost=omegaP,deltaPost=deltaP,
   gammaPost=gammaP,GammaPost=GammaP))
}
l=primoAGG(csi, omega, y, X)

evbeta=function(l){
  p=nrow(l$omegaPost)
  n=ncol(l$deltaPost)
  om=diag(sqrt(diag(l$omegaPost)))
  den=mvNcdf(l=rep(-Inf,n), u=l$gammaPost, Sig=l$GammaPost,10^4)$prob
  eta <- matrix(0,n,1)
  for (i in 1:n){
    eta[i,1] <- dnorm(l$gammaPost[i])*
                mvNcdf(l=rep(-Inf,n-1), u=l$gammaPost[-i]-
                (l$GammaPost[,i])[-i]*l$gammaPost[i],
                Sig=(l$GammaPost[,-i])[-i,]-
                (l$GammaPost[,i])[-i]%*%t((l$GammaPost[,i])[-i]),10^4)
   $prob
    print(i)}
  return(l$csiPost+om%*%l$deltaPost%*%eta)
}
```

Si riporta la funzione per ottenere l'aggiornamento dei parametri con l'algoritmo EP (Paragrafo 3.4.1) e la stima dei β con l'utilizzo di tutto il *dataset*. La versione originale dei codici è disponibile al *link*: `https://github.com/augustofasano/EPprobit-SN`

```
zeta1 = function(x){exp(dnorm(x, log = T) - pnorm(x,log.p = T))}
zeta2 = function(x,z1){-z1^2-x*z1}

getParamsEP = function(X,y,Omega,r,tolerance=1e-3,maxIter=1e3,fullVar=
    TRUE,predictive=FALSE,nPrint=100){

  n = dim(X)[1]
  p = dim(X)[2]

  invQ = Omega
  k = double(length = n)
  m = double(length = n)
  diff = 1
  nIter = 0

   while(diff > tolerance && nIter < maxIter){
      diff = 0.
      count = 0
      for(i in 1:n){
        xi = X[i,]
        r_i = r - m[i]*xi
        Oxi = invQ%*%xi
        Oi = invQ + tcrossprod(Oxi)*k[i]/as.double(1.-k[i]*crossprod(
   xi,Oxi))
        Oixi = Oi%*%xi
        xiOixi = as.double(crossprod(xi,Oixi))

        if(xiOixi>0){

          r_iOixi = as.double(crossprod(r_i,Oixi))

          s = (2.*y[i]-1.)/sqrt(1.+xiOixi)
          tau = s*r_iOixi

          z1 = zeta1(tau)
          z2 = zeta2(tau,z1)

          kNew = - z2/(1.+xiOixi+z2*xiOixi)
          mNew = s*z1 + kNew*r_iOixi + kNew*s*z1*xiOixi
```

```
          maxDiff = max(abs(c(kNew - k[i], mNew - m[i])))
          if(maxDiff>diff){diff = maxDiff}

          k[i] = kNew
          m[i] = mNew

          r = r_i + m[i]*xi

          invQ = Oi - tcrossprod(Oixi)*k[i]/(1.+k[i]*xiOixi)
        }else{
          count = count+1
          print(paste0(count," units skipped"))
        }

      }

      nIter = nIter + 1
      if(nIter %% nPrint == 0) {print(paste0("iteration ",nIter))}
    }


    meanBeta = invQ%*%r
    diagOmega = diag(invQ)


  results = list(meanBeta = meanBeta, diagOmega = diagOmega,
                 nIter = nIter, kEP = k, mEP = m, rEP=r)

  if(fullVar==TRUE){

    results = append(list(Omega=invQ),results)
  }

  if(predictive==TRUE){
      if(fullVar==FALSE){
        results = append(list(Omega=invQ),results)
      }

  }

  return(results)
}
```

```
probEP=function(Mat, d, Omega0, r0){
  y=d$y3
  fight=unique(d$date2)
  n=nrow(Mat)
  p=ncol(Mat)
  bete=matrix(NA, nrow=p, ncol=length(fight))

  for(i in 1:length(fight)){
    quali=which(d$date2==fight[i])
    Xi=Mat[quali,]
    yi=y[quali]
    get=getParamsEP(Xi, yi, Omega0, r0)
    Omega0=get$Omega
    r0=get$rEP
    bete[,i]=get$meanBeta
    cat(i,"\n")
  }
  return(bete)
}
```

Bibliografia

Agresti, A. (2015). *Foundations of Linear and Generalized Linear Models*. Wiley Series in Probability and Statistics. Wiley.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*.

Arellano-Valle, R. & Azzalini, A. (2006). On the unification of families of skew-normal distributions. *Scandinavian Journal of Statistics* **33**, 561–574.

Artuso, D. (2020). *Valutazione statistica delle performance dei giocatori della National Basketball Association*. Tesi di Laurea Magistrale, Dipartimento di Scienze Statistiche, Università degli Studi di Padova.

Azzalini, A. & Bacchieri, A. (2010). A prospective combination of phase II and phase III in drug development. *Metron - International Journal of Statistics* **68**, 347–369.

Azzalini, A. & Capitanio, A. (2014). *The Skew-Normal and Related Families*. IMS monographs. Cambridge University Press.

Azzalini, A. & Dalla Valle, A. (1996). The multivariate skew-normal distribution. *Biometrika* **83**, 715–726.

Azzalini, A. & Scarpa, B. (2012). *Data Analysis and Data Mining: An Introduction*. Oxford University Press.

BasketballReference (2018). 2018-19 NBA player stats: Advanced. `https://www.basketball-reference.com/leagues/NBA_2019_advanced.html`.

BigDataBall (2007). Dataset NBA 2018/2019. `https://www.bigdataball.com/datasets/nba/play-by-play/`.

Botev, Z. & Belzile, L. (2021). *TruncatedNormal: Truncated Multivariate Normal and Student Distributions*. R package version 2.7.2.

Botev, Z. I. (2017). The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society. Series B* **79**, 125–148.

Chopin, N. & Ridgway, J. (2017). Leave Pima Indians Alone: Binary Regression as a Benchmark for Bayesian Computation. *Statistical Science* **32**, 64 – 87.

Dandolo, D. (2019). *Valutazione statistica delle performance dei giocatori della Serie A*. Tesi di Laurea Magistrale, Dipartimento di Scienze Statistiche, Università degli Studi di Padova.

Decroos, T., Bransen, L., Van Haaren, J. & Davis, J. (2019). Actions speak louder than goals: Valuing player actions in soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Durante, D. (2019). Conjugate Bayes for probit regression via unified skew-normal distributions. *Biometrika* **106**, 765–779.

Fasano, A., Anceschi, N., Franzolini, B. & Rebaudo, G. (2023). Efficient expectation propagation for posterior approximation in high-dimensional probit models. *Book of Short Papers - SIS 2023* , 1133–1138.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80**, 27–38.

Genz, A. & Bretz, F. (2009). *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics. Springer.

Ibrahim, J., Chen, M. H., Gwon, Y. & Chen, F. (2015). The power prior: Theory and applications. *Statistics in Medicine* **34**, 3724–3749.

Ibrahim, J. G., Chen, M.-H. & Sinha, D. (2003). On optimality properties of the power prior. *Journal of the American Statistical Association* **98**, 204–213.

Jeffreys, H. S. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **186**, 453 – 461.

Kosmidis, I. (2020). *brglm2: Bias Reduction in Generalized Linear Models*. R package version 0.6.2.

Kosmidis, I. & Firth, D. (2020). Jeffreys-prior penalty, finiteness and shrinkage in binomial-response generalized linear models. *Biometrika* **108**, 71–82.

Myers, D. (2020). About box plus/minus. `https://www.basketball-reference.com/about/bpm2.html`.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Salvan, A., Sartori, N. & Pace, L. (2020). *Modelli Lineari Generalizzati*. UNITEXT. Springer.

Salvan, A., Sartori, N. & Pace, L. (2023). *Statistical Inference: Theory and Methods*. Dispensa didattica, Dipartimento di Scienze Statistiche.

Vehtari, A., Gelman, A., Sivula, T., Jylänki, P., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D. & Robert, C. P. (2020). Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *The Journal of Machine Learning Research* **21**, 577–629.

Wickham, H., François, R., Henry, L., Müller, K. & Vaughan, D. (2023). *dplyr: A Grammar of Data Manipulation*. `https://github.com/tidyverse/dplyr`.

Xianyi, Z., Kroeker, M., Saar, W., Qian, W., Chothia, Z., Shaohu, C. & Wen, L. (2023). *OpenBLAS, An optimized BLAS library*.

Zhao, Q., Sur, P. & Candès, E. J. (2020). The asymptotic distribution of the MLE in high-dimensional logistic models: Arbitrary covariance. *Bernoulli* **28**, 1835–1861.