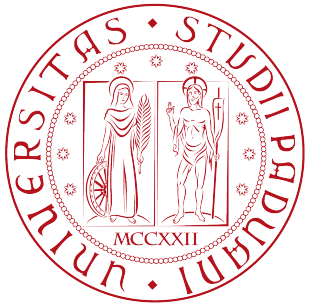
Università degli studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Triennale in

Statistica per l'Economia e l'Impresa



Relazione Finale

Modellazione statistica di risultati calcistici

Relatore Prof. Nicola Sartori
Dipartimento di Scienze Statistiche

Laureando Dandolo David
Matricola N 1100342

Anno Accademico 2016/2017

Indice

Elenco dei codici

Elenco delle tabelle

Introduzione

Nelle scommesse sportive le quote, dall'apertura alla chiusura delle puntate, continuano a cambiare sulla base del mercato, alzandosi o abbassandosi a seconda dei flussi di giocate ricevute. Le quote d'apertura (emesse circa con una settimana d'anticipo rispetto alla data dell'evento) invece vengono rilasciate su base probabilistica, dove la quota di ogni esito è uguale all'inverso della probabilità di realizzazione dell'esito stesso, ovvero $q = 1/p$. Le agenzie di scommesse per calcolare delle valide quote iniziali devono allora disporre di un modello che consenta di stimare con adeguata precisione le probabilità dei vari eventi su cui è possibile scommetere. In questa tesi studieremo appunto la costruzione di un modello adatto, non solo a questo scopo, ma che costituisca anche uno strumento per l'analisi dei risultati delle partite di calcio.
Nel Capitolo 1 parleremo brevemente della teoria della verosimiglianza, che costituisce uno strumento fondamentale dell' inferenza statistica, ovvero quell'insieme di tecniche e procedure tramite le quali quali si riesce, dall'osservazione di un campione a ricavare le caratteristiche della popolazione da cui il campione stesso si ipotizza sia stato estratto.
Nel Capitolo 2 cercheremo di trovare una distribuzione di probabilità che possa fornire una buona rappresentazione per la distribuzione del numero di gol segnati, analizzeremo quelli che sono gli aspetti da tenere in cosiderazione nella modellazione di risultati calcistici come: la differenza tra gol segnati in casa e fuori casa e quindi la presenza dell'effetto campo, la quantità di dati necessari per costruire un modello affidabile, l'influenza dello stato di forma sulle prestazioni di una squadra, e formuleremo il modello proposto da Dixon e Coles nel loro articolo del 1997 (Dixon e Coles, 1997).

Infine, nel Capitolo 3 mostreremo delle applicazioni concrete del modello, esplorando anche ambiti diversi dalla previsione di risultati, come il confronto tra squadre e/o campionati diversi, oltre che allo studio dell'andamento dinamico e non regolare delle squadre, usando dati provenienti dalla prima divisone del campionato italiano, la Serie A, e dalle prime 3 divisoni del campionato inglese, la Premier Legaue, la League Championship, e la League One. Per mostrare esempi degli utilizzi del modello, useremo il software *R* (R Core Team, 2016), e nell'appendice A riporteremo il codice prodotto necessario per l'implementazione delle tecniche e delle procedure descritte all'interno di questa tesi.

Capitolo 1

Teoria della verosimiglianza

1.1 Modello statistico

Si assume che i dati y siano una realizzazione di una variabile casuale Y con funzione di densità/probabilità nel caso continuo/discreto $f_Y(y;\theta)$ dipendente da un ignoto parametro. Indichiamo con $\mathcal{F} = \{\ f_Y(y;\theta)\ ,\ \theta \in \Theta \subseteq \mathbb{R}^p\ \}$, il modello statistico in cui θ è un parametro p-dimensionale contenuto nello spazio parametrico Θ.
Sia $f_Y^0(y)$ la vera e ignota densità di probabilità di Y, se essa è contenuta in $\mathcal{F}$, allora si dice che il modello è correttamente specificato. Se esiste una relazione biunivoca tra gli elementi di $\mathcal{F}$ e lo spazio parametrico Θ, allora il modello è detto identificabile. Ciò significa che ad ogni $\theta \in \Theta$ è associato una sola densità in $\mathcal{F}$.
Se il modello è correttamente specificato e θ è identificabile, allora il valore θ_0 di θ tale che $f_Y(y;\theta_0) = f_Y^0(y)$ è detto vero valore del parametro.

1.2 La funzione di verosimiglianza

La funzione di verosimiglianza riassume tutte le informazioni a nostra disposizione su θ. Confrontando diversi valori del parametro, cercando quello che massimizzi il valore della funzione, che costituirà quindi la miglior stima del vero valore ignoto. Inoltre, sempre tramite la verosmiglianza, è possibile

trovare regioni di confidenza per il valore del parmetro e fare verifiche d'ipotesi. La funzione $L : \Theta \to \mathbb{R}^+$, è detta funzione di verosimiglianza per θ basata sui dati y, ed è definta come:

$$L(\theta) = c(y) f_Y(y; \theta),$$

dove $c(y) > 0$ è una costante non dipendente da θ ma solo dalle realizzazioni osservate di y. Ai fini della ricerca della miglior stima del parametro, possiamo escludere $c(y)$ ed ottenere una verosimiglianza equivalente, con andamento analogo a quella originale. Con molteplici osservazioni y_i tratte da variabili indipendenti ed identicamente distribuite (*i.i.d.*), ciascuna con funzione di densità $f_Y(\theta; y_i)$, la funzione di verosimiglianza complessiva è data dal prodotto delle verosimiglianze per ogni osservazione

$$L(\theta) = \prod_{i=1}^{n} f_Y(\theta, y_i)\,.$$

Per semplicità di calcolo, spesso si utilizza la funzione di log-verosimiglianza

$$l(\theta) = \log L(\theta)\,,$$

in cui, se $L(\theta) = 0$ allora $l(\theta) = -\infty$.
La funzione di verosimiglianza gode della proprietà di invarianza che dice che $L(\theta)$ è invariante rispetto a riparametrizzazoni. Una riparametrizzaione Ψ è definta come una trasformazione biiettiva $\psi = \psi(\theta)$ con inversa $\theta = \theta(\psi)$. Un' ulteriore caratteristica che vale la pena sottolineare è che spesso non è necessario disporre di tutte le singole osservazioni sui dati per il calcolo della funzione di verosimiglianza. Una statistica $T = T(Y)$ è detta sufficiente per l'inferenza su θ se la densità di Y può essere fattorizzata in

$$f_Y(y; \theta) = f_T(t; \theta) f_{Y|T=t}(y; t)\,,$$

in cui la densità condizionata $f_{Y|T=t}$ non dipende dal valore di θ. Perciò, per costruire la verosimiglianza, ciò che è necessario, è proprio il valore della statistica sufficiente.
Inoltre, una statistica sufficente T viene definita statistica sufficiente minimale per θ se è funzione di ogni altra statistica sufficiente.

1.2.1 Condizioni di regolarità

Un modello statistico è definito regolare se rispetta le seguenti condizioni:

1. il modello è identificabile,
2. il modello è correttamente specificato,
3. lo spazio parametrico Θ è un sottoinsieme aperto di $\mathbb{R}^p$,
4. tutte le funzioni di densità specificate da $\mathcal{F}$ devono avere lo stesso supporto,
5. la funzione di log-verosimiglianza deve essere derivabile almeno fino al terzo ordine, con derivate parziali rispetto a θ continue.

1.2.2 Stima di massima verosimiglianza

Dati $\theta^{'}$ e $\theta^{''}$, è possibile confrontarli alla luce dei dati y, tramite il rapporto di verosimiglianza

$$L(\theta^{''})/L(\theta^{'})\,.$$

Se tale rapporto è maggiore di 1 si ha che $\theta^{''}$ risulta più supportato dai dati rispetto a $\theta^{'}$. In questo senso, preso un valore $\hat{\theta}$ tale che $L(\hat{\theta}) \geq L(\theta)$ per ogni $\theta \in \Theta$, allora $\hat{\theta}$ è detto stima di massima verosimiglianza di θ. In altre parole, la stima di massima verosimiglianza, è il valore di θ che massimizza la funzione di verosimiglianza. Solitamente il valore $\hat{\theta}$ può essere calcolato più agevolmente utilizzando la funzione di log-verosimiglianza. In un modello con verosimiglianza regolare (Paragrafo 1.2.1), la stima di massima verosimiglianza va cercata tra le soluzioni dell'equazione di verosimiglianza

$$l_*(\theta) = 0\,, \tag{1.1}$$

dove $l_*(\theta)$ è il vettore delle derivate parziali prime della funzione di log verosimiglianza,

$$l_*(\theta) = \left(\frac{\partial l(\theta)}{\partial \theta_1}, ..., \frac{\partial l(\theta)}{\partial \theta_p}\right)^T \tag{1.2}$$

e viene indicato con il nome di funzione punteggio, o *score*.
Solo in alcuni modelli statistici notevoli, è possibile trovare algebricamente la

soluzione dell'equazione di verosimiglianza, ma più in generale $\hat{\theta}$ va determinato numericamente, con ad esempio l'algoritmo di Newton Raphson o altri metodi per il calcolo approssimato di una soluzione di un'equazione.
La matrice $p \times p$ delle derivate parziali seconde di $l(\theta)$ cambiate di segno è detta matrice di informazione osservata, e fornisce una misura sulla curvatura locale della log-verosimiglianza

$$j(\theta) = -l_{**}(\theta) = - \begin{bmatrix} \frac{\partial^2 l(\theta)}{\partial {\theta_1}^2} & \frac{\partial^2 l(\theta)}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 l(\theta)}{\partial \theta_1 \partial \theta_p} \\ \frac{\partial^2 l(\theta)}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 l(\theta)}{\partial {\theta_2}^2} & \cdots & \frac{\partial^2 l(\theta)}{\partial \theta_2 \partial \theta_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 l(\theta)}{\partial \theta_p \partial \theta_1} & \frac{\partial^2 l(\theta)}{\partial \theta_p \partial \theta_2} & \cdots & \frac{\partial^2 l(\theta)}{\partial {\theta_p}^2} \end{bmatrix}.$$

Il valore atteso dell'informazione osservata: $i(\theta) = E_\theta(j(\theta))$ viene chiamata informazione attesa o informazione di Fisher. Nel caso di osservazioni $i.i.d$ è possibile calcolare l'informazione attesa come

$$i(\theta) = n i_1(\theta)$$

dove $i_1(\theta)$ è l'informazione attesa per una singola osservazione.

1.2.3 Proprietà campionarie

Assumendo di trovarci in presenza di un modello che soddisfi le condizioni di regolarità, valgono i seguenti risultati:

- la funzione *score* ha valore atteso pari a 0

$$E_\theta[l_*(\theta)] = 0, \qquad \forall \theta \in \Theta,$$

- vale l'identità dell'informazione, ovvero

$$E_\theta[l_*(\theta) l_*(\theta)^T] = i(\theta), \qquad \forall \theta \in \Theta,$$

  pertanto la matrice $i(\theta)$ è pari alla matrice di covarianza dello *score* e ne consegue che si tratta di una matrice definita non negativa.

1.2.4 Teoria asintotica

La variabile casuale $\hat{\theta} = \hat{\theta}(Y)$ è detta stimatore di massima verosimiglianza. Sotto condizioni di regolarità, con $y = (y_1, \ldots, y_n)$ e n sufficientemente grande, valgono alcuni risultati utili per la distribuzione (asintotica) dello stimatore di massima verosimiglianza, che risultano utili nei test e nella costruzione di intervalli di confidenza. Lo stimatore di massima verosimiglianza è consistente, ovvero assunto θ vero valore del parametro $\hat{\theta}_n \xrightarrow{p} \theta$ dove $\xrightarrow{p}$ indica la convergenza in probabilità. La convergenza di probabilità indica che

$$\lim_{n\to\infty} P(|\hat{\theta}_n - \theta| < \varepsilon) = 1, \qquad \forall \varepsilon > 0 \, .$$

Attraverso il teorema del limite centrale è possibile dimostrare che la funzione *score* segue asintoticamente una distribuzione normale di media 0 e varianza data dall'informazione attesa:

$$l_*(\theta) \dot{\sim} N_p(0, i(\theta)) \, ,$$

da cui possiamo ricavare, attraverso opportuni sviluppi in serie

$$\hat{\theta}_n - \theta \dot{\sim} N_p(0, i(\theta)^{-1}) \, .$$

In alternativa, se il calcolo di $i(\theta)$ risulta complicato, l'approssimazione è valida anche utilizzando $j(\theta)$ o $j(\hat{\theta})$

$$\hat{\theta}_n - \theta \dot{\sim} N_p(0, j(\theta)^{-1}) \, .$$

Per la proprietà della distribuzione normale si ricava

$$\hat{\theta} \dot{\sim} N_p(\theta, j(\theta)^{-1}) \, .$$

Infine, se il parametro θ ha dimensione p, allora l'i-esimo elemento di $\hat{\theta}$ ha distribuzione approssimata

$$\hat{\theta}_i \dot{\sim} N(\theta_i, [i^{-1}(\theta)]_{ii}) \, .$$

Per approfondimenti sulla teoria della verosimiglianza si vedano ad esempio Azzalini (2001) e Pace e Salvan (2001).

Capitolo 2

Il modello Dixon-Coles per risultati calcistici

2.1 La variabile casuale di Poisson

2.1.1 Il processo di Poisson

Il processo di Poisson è costituito da una serie di variabili casuali $N_t, t \geq 0$ in cui N_t può assumere i valori 0,1,2... ed esprime il numero di eventi che si realizzano nell'intervallo di tempo $[0, t)$. Indichiamo con $N(t, t+h)$ la variabile casuale che rappresenta il numero di eventi nell'intervallo $[t, t+h)$, con $t \geq 0$ e $h > 0$.

Poniamo $N(0) = 0$, ovvero indichiamo semplicemente che il conteggio degli eventi cominciato in $t = 0$ parte da 0.
I due fattori che condizionano un processo di Poisson sono:

- Si assume che le variabili $N(t, t+h)$ e N_t siano indipendenti. Questo significa che gli eventi che avvengono nei due intervalli di tempo disgiunti $[0, t)$ e $[t, t+h)$ sono indipendenti. In altre parole, questo significa che lo stato passato del processo non condiziona lo stato futuro, per questo il processo di Poisson è detto "senza memoria".

- $N_t, t \geq 0$ è detto processo di Poisson (omogeneo) se, per h sufficientemente piccolo e $v > 0$

$$Pr[N(t, t+h) = 0] = 1 - vh + o(h)$$
$$Pr[N(t, t+h) = 1] = vh + o(h)$$
$$Pr[N(t, t+h) \geq 2] = o(h)$$

  dove $o(h)$ è tale che $\lim_{h\to 0} o(h)/h = 0$.

Queste tre equazioni indicano che, in un intervallo di tempo sufficientemente piccolo il numero di avvenimenti è difficilmente maggiore di 1. Il parametro v è detto intensità del processo. Si può quindi dimostrare che la variabile casuale $N(s, s+t)$ ha una distribuzione Poisson di media vt.

2.1.2 La distribuzione di Poisson

La distribuzione di Poisson esprime, quindi, la probabilità che si verifichi un numero finito di eventi in un periodo di tempo fissato a priori con l'ipotesi che essi accadano indipendentemente l'uno dall'altro.
Data $X \sim Pois(\lambda)$, la funzione di probabilità è

$$f(x) = e^{-\lambda}\frac{\lambda}{x!}. \tag{2.1}$$

La funzione di ripartizione è

$$F(x) = \sum_{x=0}^{\infty} f(x) = \sum_{x=0}^{\infty} e^{-\lambda}\frac{\lambda}{x!}\,. \tag{2.2}$$

Si vedano le Figure 2.1 e 2.2.
Il valore medio e la varianza di X coincidono con il parametro λ, cioè

$$E(X) = Var(X) = \lambda\,.$$

Un'altra caratteristica peculiare della variabile casuale di Poisson è la sua capacità di approssimare la distribuzione Binomiale. Data una variabile casuale Binomiale, $X \sim Bin(n, p)$ se $n \to \infty$ e contemporaneamente $p \to 0$ (per cui è raro il verificarsi dell'evento), in modo che np tenda ad una costante $\lambda < 5$ allora la distribuzione della variabile casuale è approssimabile con una distribuzione Poisson di parametro λ.

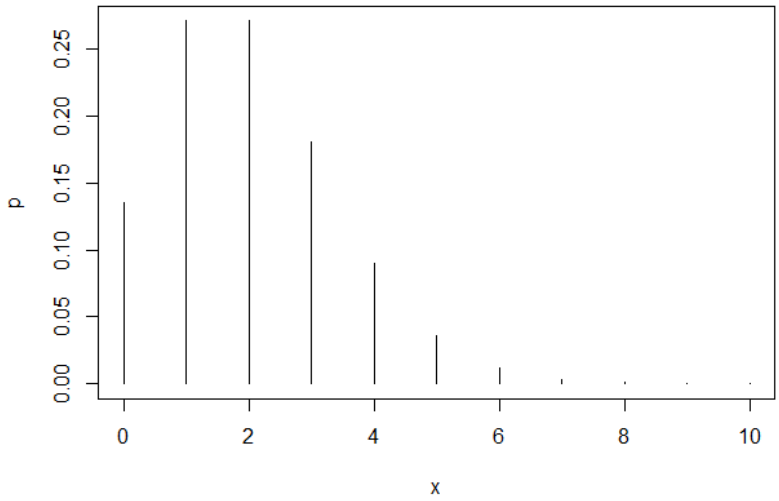


Figura 2.1: Funzione di probabilità Poisson($\lambda = 2$).

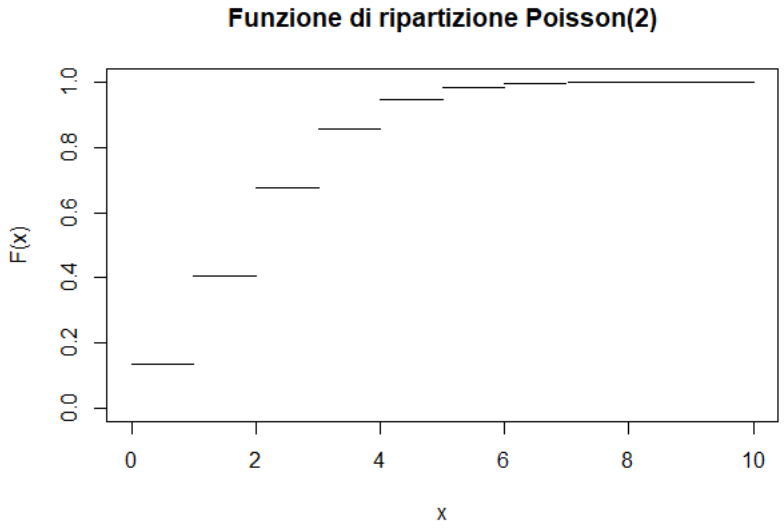


Figura 2.2: Funzione di ripartizione Poisson($\lambda = 2$).

2.1.3 La distribuzione di probabilità per il numero di gol

Ogni volta che una squadra è in possesso del pallone ha l'opportunità di attaccare e di realizzare quindi un gol. Ovviamente la probabilità p di segnare è piccola, in quanto il numero di azioni offensive che si tramutano effettivamente in una marcatura è molto basso. Se consideriamo p costante e le azioni d'attacco sono tra loro indipendenti, il numero di gol avrebbe allora una distribuzione di tipo Binomiale. In questo contesto quindi l'approssimazione della Binomiale con la Poisson risulta assolutamente adatto.

Ripensando alle caratteristiche del processo di Poisson, l'assunzione che in un piccolo intervallo di tempo la probabilità di vedere più di un gol tende a 0 sembra corrispondere perfettamente all'andamento di una partita. Tuttavia è la seconda assunzione, ovvero quella di indipendenza nei diversi intervalli di tempo, quella su cui vale la pena porsi dei dubbi. Viene naturale infatti pensare che una squadra aumenti l'intensità d'attacco se sta perdendo, o si concentri di più sulla fase difensiva se sta vincendo. O ancora si può osservare un incremento di condizione e fiducia in una squadra che ha segnato che dovrebbe aumentare le probabilità di segnare. Già nel 1951 Moroney (1951), dubitando della distribuzione di Poisson, condusse alcune analisi e propose alcune alternative, tra cui una distribuzione Poisson modificata. Tuttavia nel corso dei suoi studi dimostrò anche che, nonostante i dubbi iniziali, l'adattamento è migliore di quello che si aspettava. Nella Figura 2.3, il grafico mostra la frequenza dei gol segnati dalla squadra in casa negli incontri di Serie A della stagione 2014-15 a confronto con i dati ottenuti dalla densità di una Poisson con parmetro λ uguale alla media aritmetica dei gol segnati dalla squadra in casa. Nella Tabella 2.1, invece lo stesso confronto viene eseguito per i dati relativi alla stagione 2015-16 tramite il test χ^2 per la bontà d'adattamento, e l'alto valore del p-value ci porta a non rifiutare la distribuzione Poisson come corretta distribuzione.

```
Pearson's Chi-squared test
data: Tabella 2.1
Chi-squared = 1.6365, df = 5, p-value = 0.8968
```

Tabella 2.1: Distribuzione teorica (arrotondata all'intero più vicino) a confronto con distribuzione empirica.

Gol	0	1	2	3	4	5+
Poisson	87	128	94	46	17	5
Osservati	89	124	100	43	15	9

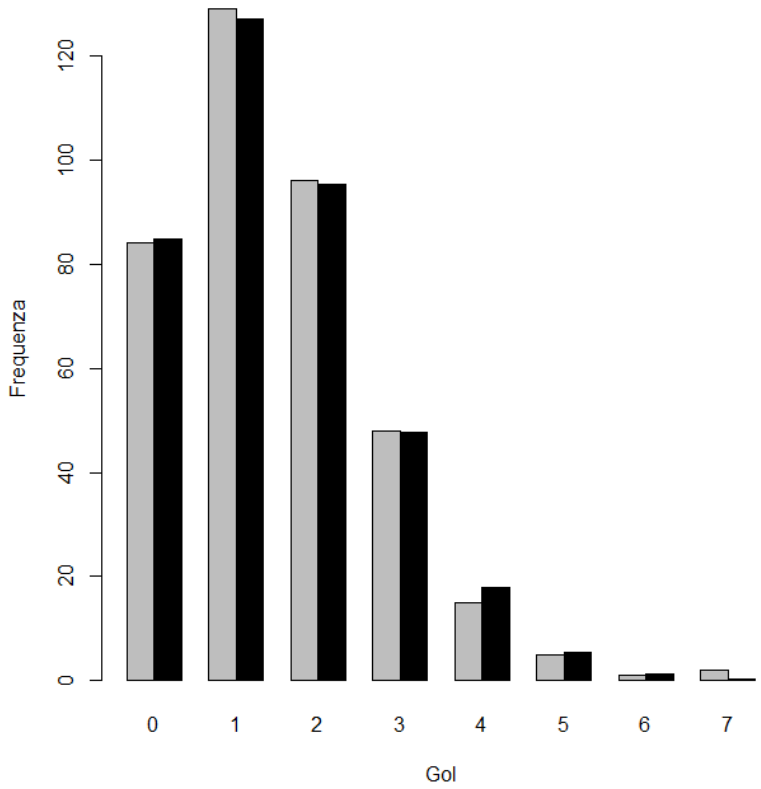


Figura 2.3: Grafico a barre del confronto tra numero di gol osservati (grigio) e teorici (nero).

Negli anni sono state proposte e adattate sui dati delle partite di vari campionati distribuzioni di diverso tipo, in particolare la Binomiale Negativa (Karlis e Ntzoufras, 2000). Uno dei motivi che porta a considerare la Binomiale Negativa è che, a differenza della distribuzione di Poisson, non vincola la varianza ad essere uguale alla media. In genere la varianza è infatti maggiore o minore, dando origine ai fenomeni di sovradispersione o sottodispersione. Nei dati riguardanti partite di calcio tuttavia, si è rilevato che la discrepanza tra media e varianza non è così marcata da indurre all'uso della Binomiale Negativa. La variabile di Poisson risulta perciò un ottimo compromesso tra semplicità di modellazione e adattamento ai dati, rappresentando di fatto una delle scelte più ricorrenti nella letteratura e negli studi statistici sui modelli per prevedere i risultati degli incontri non solo calcistici, ma sportivi in generale.

2.2 Il modello base

Come spiegato quindi, è possibile vedere il numero di gol segnati in una partita come la realizzazione di una variabile casuale Poisson, ed il risultato di una partita è dato dai gol della squadra in casa e della squadra fuori casa. Definiamo:

$$X_{ij} \sim Pois(\lambda_{i,j}) \qquad Y_{ij} \sim Pois(\mu_{i,j}), \tag{2.3}$$

dove X_{ij} rappresenta il numero dei gol segnati dalla squadra i che gioca in casa contro la squadra j, Y_{ij} i gol seganti dalla squadra j che gioca fuori casa contro la squadra i. Consideriamo le due variabili indipendenti, allora la funzione di probabilità congiunta è data dal prodotto delle densità marginali, ovvero:

$$P(X_{ij} = x, Y_{ij} = y) = P(X_{ij} = x)P(Y_{ij} = y) = \lambda_{ij}^{x}\frac{e^{-\lambda_{ij}}}{x!}\mu_{ij}^{y}\frac{e^{-\mu_{ij}}}{y!} \ . \tag{2.4}$$

La notazione (2.3) indica che la media dei gol segnati dipende da entrambe le squadre coinvolte nella partita, come viene da pensare, visto che una squadra "media" avrà buona probabilità di segnare contro una squadra considerata "debole", ma farà più fatica a farlo contro una squadra considerata "forte". Pensando ad una partita di calcio, distinguiamo genericamente due

fasi: la fase difensiva e la fase offensiva. Viene quindi intuitivo pensare che i gol segnati in una partita dalla squadra i, dipendano dall'interazione dell'abilità ad attaccare della squadra i con l'abilità di difendere della squadra j e viceversa per i gol segnati dalla squadra j. Un altro elemento da tenere in considerazione è il fatto che, in genere, si può riscontrare nella squadra che gioca in casa un vantaggio chiamato "fattore casa" o "fattore campo" (si veda ad esempio la Tabella 2.2). Molteplici fattori possono giustificare questo beneficio, come la mancanza di affaticamento per il viaggio, l'adattamento ad un terreno di gioco sintetico (dove presente), la presenza di maggior tifo, o ancora ad altri fattori fisico-psicologici (si vedano Legaz-Arrese, Moliner-Urdiales e Munguía-Izquierdo, 2013; Goumas, 2015). Nella Tabella 2.2 si può chiaramente vedere come per quasi tutte le squadre, ad eccezione di Juventus e Torino, i gol segnati in casa sono maggiori di quelli segnati fuori casa.

Tabella 2.2: Fattore Casa: Serie A 2015-16.

Squadra	Gol casa	Gol fuori casa
AC Milan	28	21
ACF Fiorentina	34	26
AS Roma	44	39
Atalanta	27	14
Bologna FC	20	13
Carpi FC	23	14
Chievo Verona	25	18
Empoli FC	22	18
Frosinone Calcio	18	17
Genoa CFC	29	16
Hellas Verona	21	13
Inter	29	21
Juventus	37	38
Lazio Roma	32	20
Sampdoria	29	19
Sassuolo Calcio	25	24
SSC Napoli	49	31
Torino FC	25	27
Udinese Calcio	18	17
US Palermo	24	14

Prendendo come base il modello di Maher (1982), Dixon e Coles (1997) hanno proposto un modello che superasse alcuni dei suoi limiti, come la possibilità di includere dati provenienti da più divisioni o da insiemi di dati non completi. Ritornando al modello (2.3), essendo la media l'unico parametro che condiziona la distribuzione di Poisson, è chiaro che tutte queste caratteristiche devono essere riassunte e contenute in essa. Perciò definiamo:

$$\log(\lambda_{ij}) = \alpha_i + \beta_j + \gamma \qquad \log(\mu_{ij}) = \alpha_j + \beta_i, \tag{2.5}$$

dove α_i e β_i indicano rispettivamente il coefficiente d'attacco e il coefficiente di difesa della squadra i-esima, e γ il fattore campo che, chiaramente, compare solo nella media della squadra in casa. Si noti che γ non è indicizzato, e quindi non è assunto diverso da squadra a squadra, ma costante. Provando ad assumere γ non costante, i risultati, in termini previsivi, non migliorano sufficentemente da giustificare l'introduzione di $n-1$ parametri aggiuntivi nel modello. La (2.5) costituisce una riparametrizzazione in scala logaritmica rispetto ai parametri proposti da Dixon e Coles, ed è utile per evitare di dover imporre $\alpha_i, \beta_i > 0$ nell'algoritmo di massimizzazione (si veda l'Appendice A.9).

2.2.1 Correzione per la non-indipendenza

Dixon e Coles, con l'uso di questo modello individuarono nei dati riguardanti 6629 incontri della Premier Laegue e delle coppe nazionali inglesi per le stagioni 1992-93, 1993-94 e 1994-95 un allontanamento dall'assunzione di indipendenza per i risultati con pochi gol (0-0,0-1,1-0,1-1). Hanno quindi proposto una modifica del modello, con una correzione di dipendenza per i risultati sopracitati, tramite l'inserimento della funzione τ (2.7) e del parametro ρ. La funzione di probabilità, tenendo conto della correzione, diventa quindi:

$$\begin{aligned} P(X_{ij} = x, Y_{ij} = y) &= P(X_{ij} = x) P(Y_{ij} = y) \tau_{\lambda,\mu}(x, y, \rho) \\ &= \lambda_{ij}^{x} \frac{e^{-\lambda_{ij}}}{x!} \mu_{ij}^{y} \frac{e^{-\mu_{ij}}}{y!} \tau_{\lambda,\mu}(x, y, \rho), \end{aligned} \tag{2.6}$$

dove

$$\tau_{\lambda,\mu}(x,y,\rho) = \begin{cases} 1-\lambda\mu\rho, & \text{se } x=y=0\,, \\ 1+\lambda\rho, & \text{se } x=0, y=1\,, \\ 1+\mu\rho, & \text{se } x=1, y=0\,, \\ 1-\rho, & \text{se } x=y=1\,, \\ 1, & \text{altrimenti} \end{cases} \tag{2.7}$$

e ρ tale che:

$$\max(-1/\lambda, -1/\mu) \leq \rho \leq \min(1/\lambda\mu, 1)\,. \tag{2.8}$$

Il parametro ρ interviene qui come un parametro di dipendenza, infatti quando $\rho = 0$ si torna al caso di indipendenza poiché la funzione τ ha come risultato in qualsiasi caso 1, mentre quando $\rho \neq 0$ e $x, y \leq 1$ allora la funzione di probabilità subisce dei cambiamenti. Nonostante questa modifica le distribuzioni marginali rimangano Poisson con media λ_{ij} e μ_{ij}. Infatti si ha

$$P(X=x) = \sum_{y=0}^{\infty} \tau_{\lambda,\mu}(x,y,\rho)\lambda^x \frac{e^{-\lambda}}{x!}\mu^y \frac{e^{-\mu}}{y!}$$

Se $x = 0$ si ha

$$\begin{aligned} P(X=0) &= e^{-\lambda}\sum_{y=0}^{\infty}\tau_{\lambda,\mu}(0,y)\frac{e^{-\mu}}{y!} \\ &= e^{-\lambda}[(1-\lambda\mu\rho)e^{-\mu} + (1+\lambda\rho)\mu e^{-\mu} + \sum_{y=2}^{\infty}\frac{e^{-\mu}}{y!}] \\ &= e^{-\lambda}[e^{-\mu} - \lambda\mu\rho e^{-\mu} + \mu e^{-\mu} + \lambda\mu\rho e^{-\mu} + \sum_{y=2}^{\infty}\frac{e^{-\mu}}{y!}] \\ &= e^{-\lambda}[\sum_{y=0}^{\infty}\frac{e^{-\mu}}{y!}] = e^{-\lambda}. \end{aligned}$$

Analogamente, se $x = 1$ si ha

$$\begin{aligned} P(X = 1) &= \lambda e^{-\lambda} \sum_{y=0}^{\infty} \tau_{\lambda,\mu}(1, y) \frac{e^{-\mu}}{y!} \\ &= \lambda e^{-\lambda} [(1 + \mu\rho)e^{-\mu} + (1 - \rho)\mu e^{-\mu} + \sum_{y=2}^{\infty} \frac{e^{-\mu}}{y!}] \\ &= \lambda e^{-\lambda} [e^{-\mu} + \mu\rho e^{-\mu} - \mu e^{-\mu} - \mu\rho e^{-\mu} + \frac{e^{-\mu}}{y!}] \\ &= \lambda e^{-\lambda} [\sum_{y=0}^{\infty} \frac{e^{-\mu}}{y!}] = \lambda e^{-\lambda}. \end{aligned}$$

Infine, per $x > 1$ si ha

$$\begin{aligned} P(X = x) &= \tau_{\lambda,\mu}(x, y, \rho) \lambda^x \frac{e^{-\lambda}}{x!} \mu^y \frac{e^{-\mu}}{y!} \\ &= \sum_{y=0}^{\infty} \lambda^x \frac{e^{-\lambda}}{x!} \mu^y \frac{e^{-\mu}}{y!} \\ &= \lambda^x \frac{e^{-\lambda}}{x!} \sum_{y=0}^{\infty} \mu^y \frac{e^{-\mu}}{y!} \\ &= \lambda^x \frac{e^{-\lambda}}{x!}. \end{aligned}$$

Notando che i risulati ottenuti con $x = 0$ e $x = 1$, coincidono proprio con la probabiltà che una Poisson di media λ sia uguale rispettivamente a 0 e 1, abbiamo dimostrato che $X \sim Pois(\lambda)$. Calcoli analoghi portano anche a dire che $Y \sim Pois(\mu)$. Nel prosieguo useremo il modello (2.6).

2.3 Stima dei parametri

Dalla (2.5) segue che il modello prevede un vettore di n parametri d'attacco $\alpha = (\alpha_1, \ldots, \alpha_n)$, un vettore di n parametri di difesa $\beta = (\beta_1, \ldots, \beta_n)$, il parametro γ per il fattore campo e il parametro di dipendenza ρ, quindi un totale di $2n + 2$ parametri. Per evitare una sovra-parametrizzazione si impone un vincolo sui parametri d'attacco:

$$\frac{1}{n} \sum_{i=1}^{n} \alpha_i = 1.$$

Un'alternativa a questo vincolo, che come si può vedere nell codice in (A.9) risulta particolarmente utile nelle stime, è data da:

$$\sum_{i=1}^{n-1} \alpha_i = -\alpha_n. \tag{2.9}$$

Le stime ottenute con un vincolo o con l'altro differiscono solo per una traslazione. Nel caso si usi il secondo vincolo possiamo infatti ottenere gli stessi coefficenti ottenuti imponendo il primo vincolo, sommando ai coefficenti d'attacco 1 e ai coefficenti di difesa -1. Per calcolare una stima del valore dei parametri usiamo la funzione di verosimiglianza:

$$L(\alpha, \beta, \rho, \gamma) = \prod_{k=1}^{K} \tau_{\lambda_k,\mu_k}(x_k, y_k) \lambda_k^{x_k} e^{-\lambda_k} \mu_k^{y_k} e^{-\mu_k}, \tag{2.10}$$

con k indice dello specifico incontro.
Le stime di massima verosimiglianza vengono calcolate tramite la massimizzazione della funzione (2.10), o della relativa log-verosmiglianza:

$$l(\alpha, \beta, \rho, \gamma) = \sum_{k=1}^{K} \log[\tau_{\lambda_k,\mu_k}(x_k, y_k)] - \lambda_k + x_k \log(\lambda_k) - \mu_k + y_k \log(\mu_k).$$

È importante fare alcune osservazioni sui dati: per poter calcolare le stime dei parametri è necessario che ci sia stato un numero di partite tale da permettere il confronto tra le squadre. Ad esempio, immaginiamo che ci siano 4 squadre: A-B-C-D; nella prima giornata A gioca con B e C gioca con D, nella seconda giornata A gioca con C e B gioca con D, nella terza A gioca con D e B gioca con C. Nella prima giornata abbiamo dati che permettono di confrontare tra loro solo la squadra A con la squadra B, e la squadra C con la squadra D. Dalla seconda giornata, nella quale abbiamo dati per confrontare anche la squadra A con la squadra C e la squadra B con la squadra D, i dati permettono un confronto tra tutte e quattro le squadre, tramite confronti incrociati che si ottengono grazie alle informazioni raccolte dalla prima giornata, nonostante non tutte le squadre abbiano già giocato tra loro, rendendo quindi possibile fare previsioni per la terza giornata. Con i dati provenienti dai campionati reali quindi, dobbiamo aspettare un numero ragionevole di

giornate dopo l'inizio della stagione, tale da permettere il confronto tra tutte le squadre.

Con dati relativi a più stagioni c'è da prestare particolare attenzione alle squadre neopromosse dalla lega inferiore o retrocesse dalla lega superiore, per le quali non abbiamo informazioni provenienti dalle stagioni precedenti. Inoltre se l'obbiettivo fosse confrontare più divisioni (ad esempio la Serie A italiana con la Serie B), ci sarebbe bisogno di usare dati pluri-stagionali per sfruttare la mobilità delle squadre dovuta a promozioni e retrocessioni. Un'altra soluzione potrebbe essere quella di includere non solo i dati provenienti dal campionato, ma anche quelli provenienti dalle partite di coppa nazionale, a cui partecipano squadre di più divisioni.

2.4 Migliorare la capacità previsiva del modello: lo stato di forma

Come in ogni sport, anche nel calcio, lo stato di forma, sia fisica che mentale, costituisce un fattore che, assieme a quelli già individuati nel Paragrafo 2.2, può influenzare l'esito delle partite. È quindi sensato pensare che le prestazioni di una squadra siano più legate ai risultati delle partite più recenti che non con quelli di partite più distanti nel tempo, ovvero assumere che i parametri del modello siano dinamici. Per fare un semplice esempio, pensiamo ad una squadra che ha giocato 6 partite, e consideriamo queste due diverse serie di risultati: V-V-P-P-S-S e P-P-S-S-V-V, dove V indica una vittoria, P un pareggio e S una sconfitta. In entrambi i casi la squadra considerata ha ottenuto 2 vittorie, 2 pareggi e 2 sconfitte, ma mentre nel primo caso la serie di risultati mostra un evidente calo di forma della squadra, nel secondo caso sembra che la squadra in oggetto si trovi in uno stato di forma ascendente. Quest'osservazione, nel tentativo di prevedere l'esito della settima partita non può essere ignorata.

Questo fenomeno è ancora più influente se i dati a nostra disposizione provengono da più stagioni. Nella finestra estiva di calciomercato avvengono i maggiori trasferimenti tra le squadre e questo porta ad una modifica nelle formazioni, che ci aspettiamo sarà riflessa nelle prestazioni e quindi dai risul-

tati. Se una squadra ha subito importanti cambiamenti nella rosa, è quindi probabile che le sue prestazioni differiscano molto da quelle dell'anno passato rendendo le informazioni che possiamo ricavere dai dati meno recenti poco significative.

Se fatto con superficialità, questo ragionamento potrebbe condurci ad escludere l'uso dei dati provenienti dalle stagioni passate, e interessarci solo a quelli relativi alla stagione per la quale vogliamo fare previsioni. Con un approccio di questo tipo corriamo però i rischi connessi al basarci su pochi dati, che possono non riflettere adeguatamente le vere abiltà di una squadra nel suo complesso. Un esempio è quello della Juventus nella stagione 2014-15, che nelle prime dieci giornate si trovava nella seconda metà della classifica, e che ha invece terminato il campionato al primo posto.

Fortunamente nel calcio, si possono individuare un fattore economico e una mentalità societaria che tendono a standardizzare l'andamento di una squadra, facendola appartenere ad un certo gruppo di posizioni per diversi anni, rendendo genericamente l'evoluzione di una squadra, come ad esempio quella che ha caratterizzato il Napoli dal 2004 dopo il fallimento, o l'involuzione, come quella subita dal Milan dal 2010 ad oggi, processi lenti, dandoci la possibiltà di includere nel modello molti più dati, provenienti da diverse stagioni, adattandone però l'importanza. La strategia quindi è di utilizzare tutti i dati, tuttavia pesando nella costruzione del modello i risultati meno recenti in modo inferiore rispetto ai risulati più recenti.

Dixon e Coles (1997) propongono una modifica all'equazione (2.10), che permette di includere nelle informazioni che vengono inserite nella funzione di verosimiglianza anche le informazioni relative alla data in cui vengono giocate le partite.

Questa modifica porta a costruire una funzione di pseudo-verosimiglianza:

$$L(\alpha, \beta, \rho, \gamma) = \prod_{k \in P_t} \{\tau_{\lambda_k, \mu_k}(x_k, y_k) \lambda_k^{x_k} e^{-\lambda_k} \mu_k^{y_k} e^{-\mu_k}\}^{\phi(t-t_k)} \tag{2.11}$$

e la corrispondente log-verosimiglianza

$$\begin{aligned} l(\alpha, \beta, \rho, \gamma) = \sum_{k \in P_t} \phi(t - t_k) \{ & \log[\tau_{\lambda_k, \mu_k}(x_k, y_k)] - \lambda_k + x_k \log(\lambda_k) \\ & - \mu_k + y_k \log(\mu_k)\}; \end{aligned}$$

dove t_k rappresenta la data in cui è stata giocata la k-esima partita, $P_t = \{k : t_k < t\}$, λ_k e μ_k sono definiti come nella (2.5) e $\phi(t)$ è una funzione decrescente.

Sotto questa prospettiva, le stime dei parametri α, β, γ, e ρ, sono a loro volta dipendenti dal tempo, dato che la data di riferimento t modifica l'insieme P_t dei dati che entrano nella verosimiglianza. Massimizzare la funzione (2.11) alla data t, porta a trovare le stime di massima verosimiglianza dei parametri relative a quella precisa data, e valutando la serie delle stime per diverse date di riferimento è possibile avere una sorta di serie storica dell'andamento delle squadre. Per la (2.11) tuttavia non essendo più una verosimiglianza propria ma una pseudo verosimiglianza, la teoria asintotica vista nel Paragrafo 1.2.4 non è più valida, in particolar modo la varianza della distribuzione asintotica dello stimatore non è più $j(\theta)^{-1}$, che andrebbe opportunamente aggiustata.
Variando la scelta della funzione $\phi(t)$, inoltre, possiamo pesare in modo diverso nella funzione di verosimiglianza le partite più lontane nel tempo.

Nel loro articolo Dixon e Coles, propongono alcune funzioni. La più semplice è

$$\phi(t) = \begin{cases} 1, & t < t_0 \\ 0, & t \geq t_0, \end{cases}$$

grazie alla quale saranno inclusi nella funzione di verosimiglianza solo gli incontri per cui $t - t_k$ sia minore di una distanza massima fissata t_0. Tuttavia i dati entranti vengono pesati tutti allo stesso modo e questo stona con le considerazioni che abbiamo fatto in precedenza.
Un'alternativa proposta invece definisce

$$\phi(t) = \exp(-\xi t)\,, \tag{2.12}$$

che fa includere, quindi, nella funzione di verosimiglianza tutte le partite precedenti alla data di riferimento, pesate in modo diverso sulla base della distanza temporale $t - t_k$ ponderata per il parametro ξ. Si noti che quando $\xi = 0$, si realizza il modello statico (2.10), quindi di fatto questa seconda proposta ne costituisce una generalizzazione. Come si può vedere nella Figura 2.4, più è grande il valore di ξ, minor peso sarà dato agli incontri più lontani.

Nei grafici si possono vedere dei piccoli crolli intorno a $t = 400$ che sono dovuti alla distanza (in giorni) tra l'ultima partita di una stagione e la prima della stagione successiva. Le irregolarità invece visibili intorno a $t = 210$ e $t = 620$ sono dovute ai recuperi delle partite rinviate a causa di supercoppe e condizioni climatiche avverse. Nei dati presi in esame infatti le partite sono ordinate per giornate di campionato e non per effettiva data di gioco, mentre noi siamo interessati ai riultati più recenti in termini di tempo.

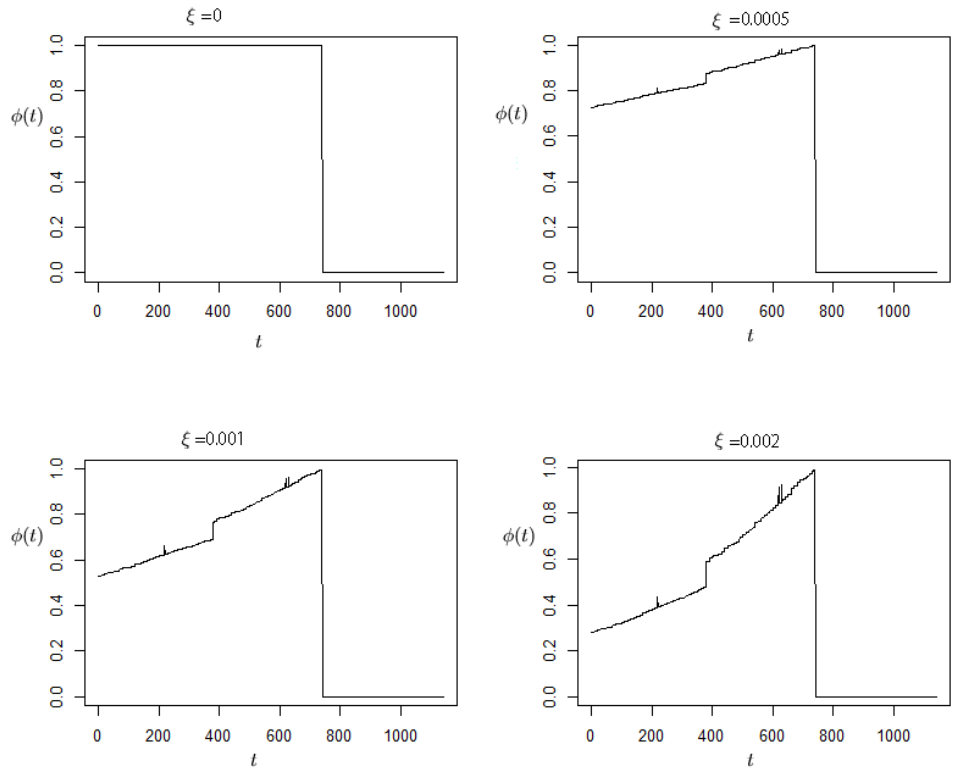


Figura 2.4: Grafico di $\phi(t)$ (2.12) per diversi valori di ξ; t è il numero di giorni di differenza tra la data di riferimento e la partita k-esima.

La scelta del parametro ξ risulta complicata, in quanto la (2.11) definisce una sequenza temporale di verosimiglianze non indipendenti, nelle quali abbiamo bisogno di un ξ tale da massimizzare la capacità previsiva del model-

lo. È quindi questo il criterio che ci dovrebbe guidare nella scelta del miglior valore per il parametro.

Procediamo con un approccio simile a quello del metodo di convalida incrociata (*cross-validation*). Dividiamo i dati, usando quelli fino alla data t_0 per stimare il modello con diversi valori di ξ, e per ognuno stimiamo le probabilità dei risultati delle partite che si giocheranno nelle, circa, due o tre giornate successive. Aggiorniamo la data t_0 e ripetiamo la procedura fino a completare le previsioni per tutte le partite a nostra disposizione. Infine, confrontiamo le stime ottenute con i risultati noti, e scegliamo il valore ξ che garantisce le migliori previsioni. Per valutare l'accuratezza delle previsioni, prima di tutto vengono calcolate le probabilità dei risultati, intesi come vittoria, pareggio, sconfitta, senza considerare il numero di gol segnati:

$$\begin{aligned} p_k^H &= \sum_{l,m \in B_H} P(X_k = l, Y_k = m)\,, \\ p_k^D &= \sum_{l,m \in B_D} P(X_k = l, Y_k = m)\,, \\ p_k^A &= \sum_{l,m \in B_A} P(X_k = l, Y_k = m)\,, \end{aligned} \tag{2.13}$$

dove $B_H = \{(l,m) : l > m\}$ rappresenta l'insieme dei risultati che corrispondono ad una vittoria della squadra in casa, $B_D = \{(l,m) : l = m\}$ ad un pareggio e $B_H = \{(l,m) : l < m\}$ ad una vittoria della squadra fuori casa. Per valutare l'affidabilità complessiva di queste stime usiamo la quantità:

$$S(\xi) = \sum_{k=1}^{N} [\theta_k^H \, log\, p_k^H + \theta_k^A \, log\, p_k^A + \theta_k^D \, log\, p_k^D], \tag{2.14}$$

dove:

- $\theta_k^H = 1$ se nella partita k-esima c'è stata una vittoria per la squadra in casa, e 0 negli altri casi,
- $\theta_k^D = 1$ se nella partita k-esima c'è stata un pareggio, e 0 negli altri casi,
- $\theta_k^A = 1$ se nella partita k-esima c'è stata una vittoria per la squadra ospite, e 0 negli altri casi.

Considerando solo i risultati, e non i gol segnati, (2.14) costituisce l'analogo di una verosimiglianza profilo predittiva, (Dixon e Coles, 1997, Paragrafo 4.4). $S(\xi)$ è una funzione crescente rispetto all'affidabilità delle stime, ed ha massimo in 0, che si verifica se $p_k^H = \theta_k^H, p_k^A = \theta_k^A, p_k^D = \theta_k^D \ \ \forall k \in K$, ovvero quando il modello stima perfettamente l'esito di ogni partita. Il valore del parametro ξ che massimizza $S(\xi)$ è quindi quello che garantisce la migliore capacità previsiva al modello.

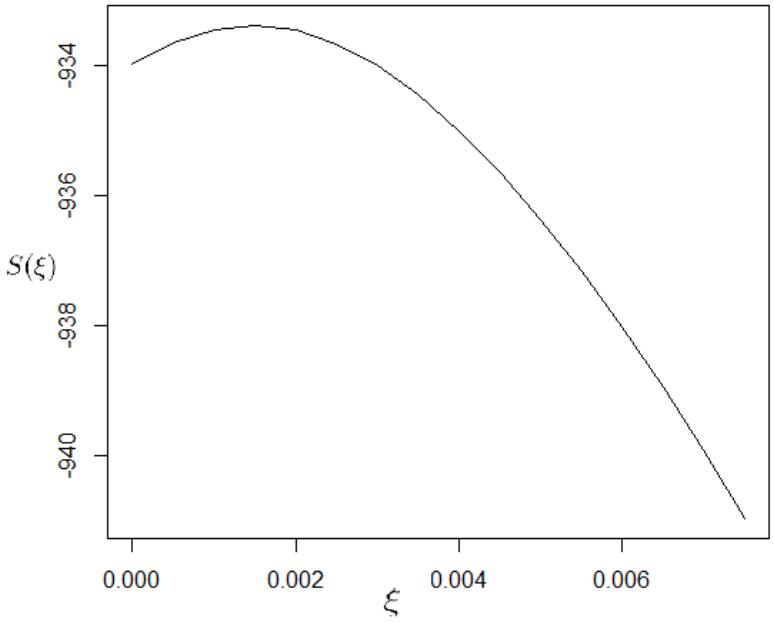


Figura 2.5: Grafico di $S(\xi)$ per i dati della Premier League 2012-13, 2013-14, 2014-15, 2015-16.

Il valore trovato nel grafico in Figura 2.5, $\xi = 0.0015$ relativo ai dati della Premier League 2012-13, 2013-14, 2014-15, 2015-16 è un valore relativamente alto, in quanto come si può vedere nella Figura 2.4 porta a pesare le partite più distanti di un anno circa 0.4, ovvero ritiene inaffidabile i dati provenienti dalle stagioni passate, facendo pensare ad un alta variabilità delle prestazioni delle squadre. Questa ipotesi è confermata dalle classifiche finali delle diverse stagioni, che mostrano eterogenenità nei piazzamenti delle squadre.

Capitolo 3

Utilizzi del modello

Nel capitolo precedente, abbiamo analizzato il modello di Dixon e Coles ponendo particolare attenzione alla sua capacità previsiva. Tuttavia la previsione dei risultati non è l'unico ambito che il modello ci consente di studiare. Grazie alle stime dei parametri d'attacco α_i e i parametri di difesa β_i possiamo:

- valutare e confrontare le squadre;
- confrontare diversi campionati e diverse divisioni;
- valutare l'andamento dinamico nel tempo delle squadre;
- prevedere risultati e simulare l'andamento di un campionato o torneo.

3.1 Valutazione e confronto delle squadre

Consideriamo i dat provenienti dalla Serie A italiana, stagioni 2012-13, 2013-14, 2014-15, 2015-16, 2016-17. Come spiegato nel Paragrafo 2.4, visto che stiamo usando dati provenienti da 4 stagioni, è il caso di utilizzare il modello (2.11) con data di riferimento quella dell'ultima partita giocata (03/04/2017). Per prima cosa quindi, determiniamo il valore ottimale di ξ, e successivamente calcoliamio le stime di massima verosimiglianza (si veda Figura 3.3). I risultati ottenuti sono riportati nella Tabella 3.1.

Home Advantage Coefficent (γ): 0.2546
Correlation Parameter (ρ): -0.0529
ξ: 0.0002

Squadra	$\hat{\alpha}$	$\hat{\beta}$	Squadra	$\hat{\alpha}$	$\hat{\beta}$
Atalanta	0.98	-0.96	Lazio	1.24	-1.06
Bologna	0.74	-0.94	Livorno	0.87	-0.56
Cagliari	0.97	-0.75	Milan	1.20	-1.08
Carpi	0.81	-0.84	Napoli	1.52	-1.14
Catania	0.92	-0.86	Palermo	0.87	-0.74
Cesena	0.78	-0.62	Parma	0.99	-0.86
Chievo	0.79	-1.00	Pescara	0.69	-0.45
Crotone	0.54	-0.72	Roma	1.45	-1.25
Empoli	0.79	-0.91	Sampdoria	1.02	-0.92
Fiorentina	1.32	-1.07	Sassuolo	1.04	-0.84
Frosinone	0.76	-0.56	Siena	0.77	-0.84
Genoa	0.98	-0.94	Torino	1.19	-0.90
Inter	1.25	-1.08	Udinese	1.00	-0.90
Juventus	1.46	-1.73	Verona	1.06	-0.71

Tabella 3.1: Stime di massima verosimiglianza dei coefficenti d'attacco e di difesa per le squadre cha hanno partecipato ai campionati di Serie A 2012-13, 2013-14, 2014-15, 2015-16, 2016-17 (30 giornate).

Il valore piccolo di ξ significa che il modello considera importanti i dati provenienti anche dalle stagioni passate (si veda la Figura 3.2), sottintendendo che nelle cinque stagioni considerate le squadre abbiano avuto un rendimento costante e comparabile. In effetti, da una rapida visione delle varie classifiche finali salta subito all'occhio che le squadre si sono collocate all'incirca nella stessa zona.

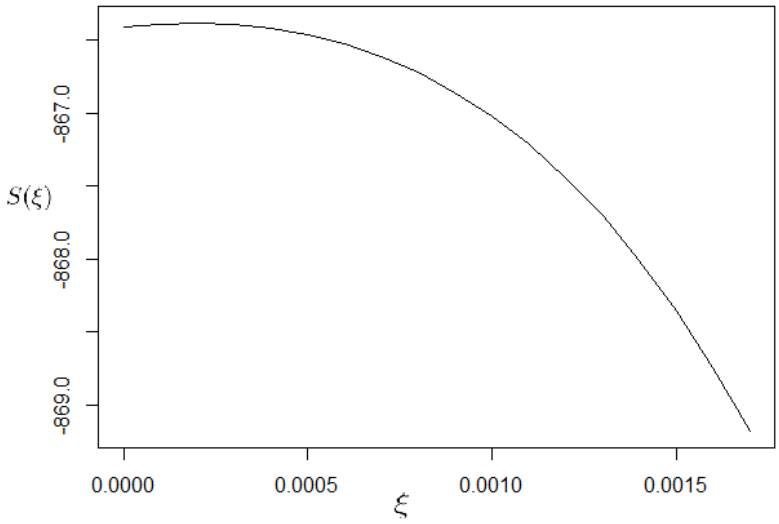


Figura 3.1: S(ξ) calcolata per la Serie A 2012-13, 2013-14, 2014-15, 2015-16, 2016-17 (30 giornate).

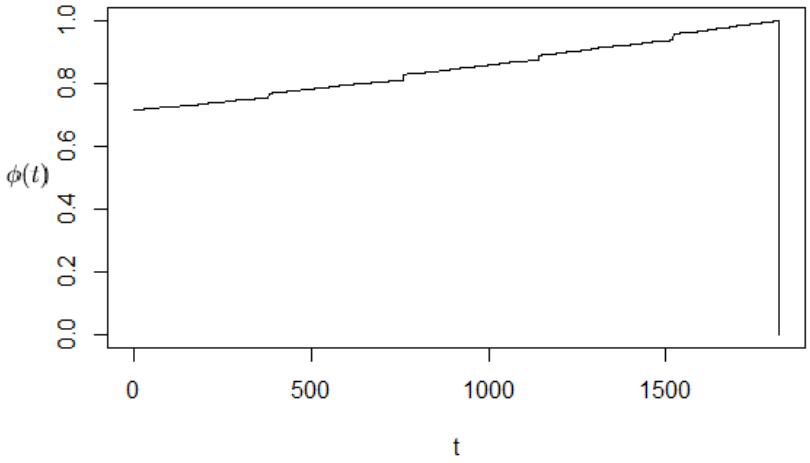


Figura 3.2: Andamento della funzione $\phi(t)$ con ξ=0.0002.

Da questi risultati possiamo procedere ad una valutazione delle abilità delle squadre. Essendo λ_{ij} crescente rispetto a α_i, β_j e γ, allora lo è anche il numero medio di gol segnati dalla squadra in casa, e ne deriva che:

- un elevato valore di α_i corrisponde una maggiore abiltà offensiva: α_i positivo fa quindi aumentare il numero medio di gol segnati;
- un basso valore di β_j corrisponde ad una maggior abiltà difensiva: β_j negativo fa diminuire il numero medio di gol segnati, riducendo la forza d'attacco della squadra avversaria;
- γ risulta sempre positivo, in quanto il fattore campo, come osservato in precedenza, aumenta il numero medio di gol segnati dalla squadra in casa.

Un ragionamento analogo vale con α_j e β_i per μ_{ij}, con l'ovvia esclusione dell'effetto campo.

Seguendo gli stessi criteri, possiamo confrontare tra loro le squadre. Prese due generiche squadre h e k, diremo che:

- h ha una forza offensiva maggiore di quella di k se $\alpha_h > \alpha_k$;
- h ha un forza difensiva maggiore di quella di k se $\beta_h < \beta_k$.

Per una visione d'insieme il grafico a barre della Figura 3.3 risulta particolarmente adatto. Possiamo vedere come la Juventus, che ha dominato gli ultimi campionati, non sia nettamente il miglior attacco, anzi è seconda dopo il Napoli ed è alla pari con Roma, tuttavia presenta una difesa nettamente superiore a tutte le altre squadre, confermando l'idea che in Italia vinca il campionato la squadra con la miglior difesa.

Figura 3.3: Grafico comparativo dei valori stimati dei coefficenti α_i (nero) e β_i (grigio) per la Serie A.

3.2 Confronto tra divisioni e campionati

Usando i dati di più divisoni dello stesso paese, è possibile valutare la differenza media tra le squadre partecipanti a diverse divisioni, grazie ai confronti incrociati consentiti dalla mobilità delle squadre, dovuta a retrocessioni e promozioni, e agli incontri di Coppa Nazionale. Per confrontare diversi campionati europei tra loro, invece dobbiamo trovare dei dati da aggiungere a quelli provenienti dai signoli campionati, che facciano da collante tra un campionato e l'altro, garantendo la confrontabilità. Per questo scopo allora i risultati degli incontri dalle coppe europee, ovvero la Champions League e l'Europa League, sembrano particolarmente adatti. Per semplicità di reperimento, usiamo i dati provenienti dalla prima e dalla seconda divisone inglese per le stagioni 2012-13, 2013-14, 2014-15 contenuti nel pacchetto R `engsoccerdata` (Curley, 2016). Applichiamo il modello e otteniamo le stime di massima verosimiglianza, che per praticità, riportiamo già divise per campionati, nelle Tabelle 3.2, 3.3, 3.4.

Home Advantage Coefficent (γ): 0.2174
Correlation Parameter (ρ): -0.0370
ξ: 0.0015

Squadra	$\hat{\alpha}$	$\hat{\beta}$	Squadra	$\hat{\alpha}$	$\hat{\beta}$
AFC Bournemouth	1.34	-1.00	Newcastle United	1.26	-1.04
Arsenal	1.70	-1.56	Norwich City	1.24	-1.05
Aston Villa	0.98	-0.97	Southampton	1.52	-1.45
Chelsea	1.68	-1.45	Stoke City	1.27	-1.25
Crystal Palace	1.21	-1.24	Sunderland	1.24	-1.15
Everton	1.50	-1.27	Swansea City	1.32	-1.25
Leicester City	1.51	-1.39	Tottenham Hotspur	1.65	-1.38
Liverpool	1.71	-1.27	Watford	1.34	-1.11
Manchester City	1.85	-1.47	West Bromwich Albion	1.14	-1.26
Manchester United	1.54	-1.55	West Ham United	1.47	-1.25

Tabella 3.2: Stime di massima verosimiglianza per le squadre partecipanti alla Premier League 2015-16.

Squadra	$\hat{\alpha}$	$\hat{\beta}$	Squadra	$\hat{\alpha}$	$\hat{\beta}$
Birmingham City	0.99	-0.92	Hull City	1.15	-1.29
Blackburn Rovers	1.02	-1.01	Ipswich Town	1.04	-1.04
Bolton Wanderers	0.92	-0.73	Leeds United	0.94	-0.89
Brentford	1.20	-0.86	Middlesbrough	1.13	-1.32
Brighton & Hove Albion	1.11	-1.19	Milton Keynes Dons	0.95	-0.77
Bristol City	1.09	-0.77	Nottingham Forest	1.02	-0.94
Burnley	1.18	-1.32	Preston North End	0.90	-1.08
Cardiff City	1.04	-0.98	Queens Park Rangers	1.05	-1.00
Charlton Athletic	0.84	-0.76	Reading	0.98	-0.87
Derby County	1.28	-1.07	Rotherham United	0.98	-0.76
Fulham	1.17	-0.67	Sheffield Wednesday	1.05	-1.07
Huddersfield Town	1.06	-0.73	Wolverhampton Wanderers	1.09	-0.97

Tabella 3.3: Stime di massima verosimiglianza per le squadre partecipanti alla Championship 2015-16.

Squadra	$\hat{\alpha}$	$\hat{\beta}$	Squadra	$\hat{\alpha}$	$\hat{\beta}$
Barnsley	0.88	-0.71	Millwall	0.89	-0.77
Blackpool	0.52	-0.61	Oldham Athletic	0.53	-0.63
Bradford City	0.68	-0.90	Peterborough United	0.96	-0.55
Burton Albion	0.70	-1.10	Port Vale	0.71	-0.58
Bury	0.65	-0.42	Rochdale	0.92	-0.55
Chesterfield	0.80	-0.56	Scunthorpe United	0.77	-0.65
Colchester United	0.71	-0.29	Sheffield United	0.78	-0.75
Coventry City	0.79	-0.67	Shrewsbury Town	0.69	-0.40
Crewe Alexandra	0.55	-0.37	Southend United	0.74	-0.52
Doncaster Rovers	0.64	-0.62	Swindon Town	0.88	-0.58
Fleetwood Town	0.59	-0.72	Walsall	0.79	-0.76
Gillingham	0.88	-0.57	Wigan Athletic	1.01	-0.90

Tabella 3.4: Stime di massima verosimiglianza per le squadre partecipanti alla League One 2015-16.

	$\hat{\alpha}$ medio	$\hat{\beta}$ medio
Premier League	1.42	-1.27
Championship	1.05	-0.96
League One	0.75	-0.63

Tabella 3.5: Valori medi dei parametri d'attaco e difesa per le squadre di ogni divisione.

I valori medi delle stime per divisioni sono riportate nella Tabella 3.5. Come ci attendevamo, il valore medio dei paramentri d'attacco è maggiore nelle divisioni più alte, mentre l'abilità difensiva si avvicina a 0 nelle divisioni più basse. All'aumentare dello scarto tra la media dei valori, e quindi della differenza di abilità media tra le squadre di diverse divisioni, ci aspettiamo una crescente difficoltà da parte delle squadre che vengono promosse a restare nella lega maggiore, e saranno quindi in genere proprio le neopromosse a retrocedere a fine campionato, mentre al diminuire di questo scarto ci aspettiamo di vedere che le squadre che vengono retrocesse siano spesso diverse.

3.3 Rendimento dinamico nel tempo

Come già discusso nel Paragrafo 2.4, il rendimento delle squadre non è costante nel tempo. Grazie al modello (2.11), usando diverse date di riferimento, possiamo costruire una sorta di serie storica di stime per i parametri (Figura 3.5), che rifletta l'andamento dinamico delle squadre, e che può essere usata per individuare un'eventuale ciclicità o stagionalità, o ancora un trend crescente, decrescente o un andamento tendenzialmente costante. Un osservazione da fare è che, nonstante anche le stime del parametro γ relativo al fattore campo cambiano nel tempo (Figura 3.4), esse rimangono circa costanti. Riflettendo, sembra infatti illogico assumere che il vantaggio dato dal giocare in casa cambi in base al periodo considerato.

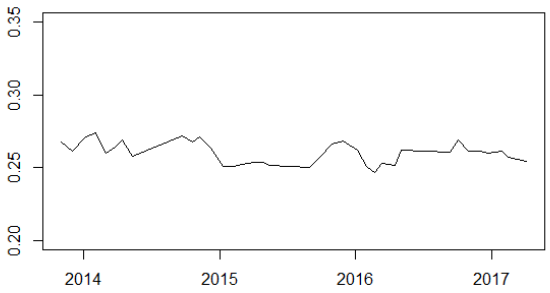


Figura 3.4: Serie storica delle stime per γ.

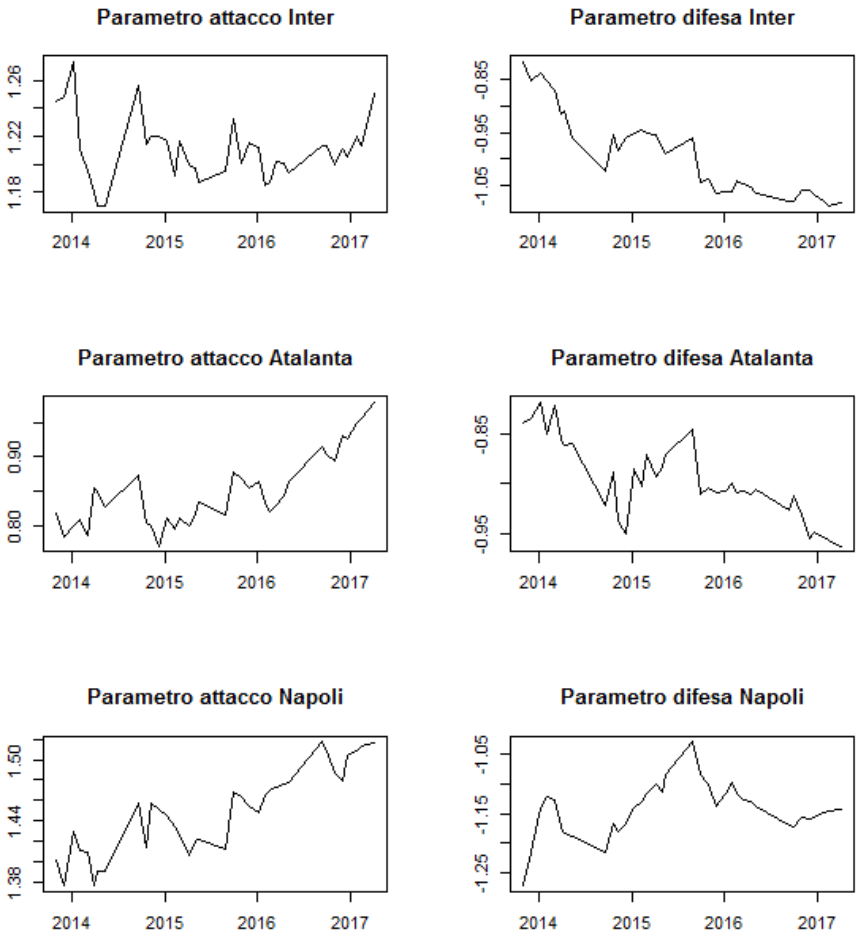


Figura 3.5: Serie storica delle stime per α e β di Inter, Atalanta, Napoli.

Nella Figura 3.5 possiamo notare come l'Inter abbia mantenuto il parametro d'attacco intorno all'1.20 (anche se con grande irregolarità), mentre ha migliorato notevomente il parametro di difesa. È evidente invece la crescita dell'Atalanta, che ha migliorato nel tempo entrambi i parametri. Il Napoli invece ha avuto un netto miglioramento nel coefficente d'attacco, ma ha anche peggiorato quello di difesa.

3.4 Previsioni e simulazioni

3.4.1 Stima delle probabilità dei risultati

Fissati $X = x$, $Y = y$ per $x, y = 0, 1, 2, 3, 4, 5, 6, ...$, dalla funzione di probabilità (2.6), possiamo calcolare una stima della probabilità di ogni risultato e, con queste, tramite le formule (2.13) possiamo calcolare la stima delle probabilità per gli esiti finali (V-P-S). Ad esempio, considerando l'incontro Milan-Atalanta, in Tabella 3.7 si hanno le probabilità dei vari risultati (fino al 4-4). Invece in Tabella 3.6 (e in Figura 3.6) vengono rappresentate le probabilità di vittoria del Milan, di pareggio e di vittoria dell'Atalanta.

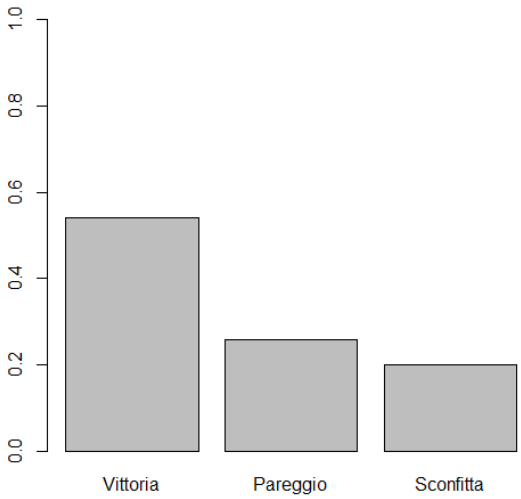


Figura 3.6: Grafico a barre delle probabilità dei risultati V-P-S per il Milan.

Vittoria (casa)	Pareggio	Sconfitta (casa)
0.46088	0.28019	0.2589

Tabella 3.6: Stima delle probabilità del risultato finale per l'incontro Milan-Atalanta.

$\lambda_{Mil,Ata}$= 1.409921 $\mu_{Mil,Ata}$ = 0.9903228

Milan	Atalanta	$\hat{p}$	Milan	Atalanta	$\hat{p}$
0	0	0.0963	3	0	0.0424
0	1	0.0842	3	1	0.0420
0	2	0.0445	3	2	0.0208
0	3	0.0147	3	3	0.0069
0	4	0.0036	3	4	0.0017
1	0	0.1223	4	0	0.0149
1	1	0.1322	4	1	0.0148
1	2	0.0627	4	2	0.0073
1	3	0.0207	4	3	0.0024
1	4	0.0051	4	4	0.0006
2	0	0.0901			
2	1	0.0893			
2	2	0.0442			
2	3	0.0146			
2	4	0.0036			

Tabella 3.7: Singoli risultati (fino al 4-4) e relativa probabilità stimata. Il risultato con la probabilità stimata più alta (0.1322) di realizzazione è 1-1.

3.4.2 Simulazioni

Simulare un processo significa replicarlo un numero R di volte molto grande, rilevandone di volta in volta lo stato finale, dando la possibilità di stimare la probabilità di un esito s del processo, come $\hat{p} = m/R$, dove m rappresenta il numero di volte in cui si è verificato l'esito s. Tecniche di questo tipo, dette Monte Carlo, vengono usate per determinare la distribuzione di variabili casuali ignote (si vedano ad esempio Chiodi, 2000 e Rubinstein e Kroese, 1981).

Per simulare un campionato intero o una frazione di esso allora, dobbiamo ripetere l'insieme delle partite che ci interessano e valutare il totale dei punti raccolti dalle squadre e la loro posizione finale un numero R di volte. In questo modo possiamo calcolare una stima della probabilità che ha ogni squadra di piazzarsi in certe zone della classifica, come ad esempio la zona di qualificazione per le coppe europee, la testa della classifica, o la zona retrocessione. Premessa per fare ciò è quella di generare risultati casuali per ogni partita, a partire dal modello stimato, con il seguente procedimento:

1) per ogni k-esima partita stimiamo la distribuzione di probabilità dei possibili esiti (come fatto nel Paragrafo 3.4.1 per la partita Milan-Atalanta),

2) per ogni k-esima partita estraiamo R volte un risultato tramite la tecnica del campionamento probabilistico, ovvero eseguiamo un' estrazione casuale in cui ogni possibile risultato ha una probabilità di estrazione pari a quella stimata al punto 1.

Infine, assegniamo ad ogni squadra la media dei punti realizzati nelle R simulazioni e se, come nel caso che vedremo in seguito, si tratta di simulazione parziale li sommiamo ai punti già realizzati.

	Squadra	Punti		Squadra	Punti
1	Juventus	74	11	Chievo	38
2	Roma	68	12	Udinese	37
3	Napoli	64	13	Cagliari	35
4	Lazio	60	14	Bologna	34
5	Atalanta	58	15	Sassuolo	31
6	Inter	55	16	Genoa	29
7	Milan	54	17	Empoli	22
8	Fiorentina	51	18	Crotone	17
9	Sampdoria	44	19	Palermo	15
10	Torino	41	20	Pescara	13

Tabella 3.8: Classifica 30-esima giornata Serie A 2016-17.

Vogliamo analizzare i dati della Serie A provenienti dalle stagioni 2012-13, 2013-14, 2014-15, 2015-16 e i dati parziali della stagione 2016-17, fino alla 30-esima giornata, con l'obbiettivo di prevedere la classifica finale. Alla data dell'elaborazione (05/04/2017) la classifica è quella riportata nella Tabella 3.8.

Squadra	Vittoria	Champions Lg	Europa Lg	Retrocessione
Atalanta	0	31	756	0
Bologna	0	0	0	0
Cagliari	0	0	0	0
Chievo	0	0	0	0
Crotone	0	0	0	964
Empoli	0	0	0	81
Fiorentina	0	0	218	0
Genoa	0	0	0	1
Inter	0	16	641	0
Juventus	968	1000	0	0
Lazio	0	118	824	0
Milan	0	4	394	0
Napoli	1	854	144	0
Palermo	0	0	0	960
Pescara	0	0	0	994
Roma	31	977	23	0
Sampdoria	0	0	0	0
Sassuolo	0	0	0	0
Torino	0	0	0	0
Udinese	0	0	0	0

Tabella 3.9: Tabella con gli stati finali ottenuti in ogniuna delle $R = 1000$ simulazioni.

Nella Tabella 3.9 possiamo vedere quante volte nelle $R = 1000$ simulazioni effettuate ogni squadra abbia raggiunto dei piazzamenti rilevanti e, dividendo per 1000, possiamo stimare le probabilità di quei piazzamenti. Dobbiamo ricordare che sono state simulate solo le ultime 8 giornate, con un massimo quindi di 24 punti disponibili per squadra, ed è quindi naturale che i le posizioni finali si adattino in parte alla classifica attuale. Per quanto riguarda il primo posto non sembrano esserci dubbi, con la Juventus che ha una pro-

Squadra	Punti	Squadra	Punti
Juventus	92	Udinese	47
Roma	84	Chievo	46
Napoli	79	Cagliari	44
Lazio	73	Bologna	42
Atalanta	69	Sassuolo	40
Inter	68	Genoa	36
Milan	67	Empoli	31
Fiorentina	64	Palermo	23
Sampdoria	54	Crotone	22
Torino	52	Pescara	19

Tabella 3.10: La classifica finale della Serie A 2016-17 stimata con la tecnica Monte Carlo.

babilità del 96,8% di vincere il campionato. Per l'ingresso nella Champions League invece oltre alla presenza certa della Juventus (100%), la Roma e/o il Napoli (rispettivamente 97,7% e 85,4%) verrano difficilmente superate da altre pretendenti come la Lazio (11,8%) e l'Atalanta (0,0031%). Per la zona di accesso all'Europa League invece sembrano avere ottime probabilità Atalanta, Inter e Lazio (rispettivamente 75,6%, 64,1% e 82,4%). Una buona probabilità anche per Milan (39,4%) e Fiorentina (21,8%), così come per il Napoli (14,4%) nell'eventualità in cui venga superato dalla Lazio in classifica. Infine per la zona retrocessione sembra già tutto deciso con Crotone, Palermo, e Pescara, che sono sempre retrocesse tranne in 82 simulazioni, di cui 81 ai danni dell'Empoli e una sola volta superando il Genoa. La Tabella 3.10 mostra invece la classifica finale stimata sommando ai punti della Tabella 3.8 la media dei punti ottenuti nelle 1000 simulazioni. Possiamo vedere come le posizioni siano rimaste quasi inalterate, con l'eccezione dell'Udinese che ha superato il Chievo. A conferma di ciò che abbiamo visto in precedenza, vediamo che la Juventus ha mantenuto un buon distacco dalla Roma seconda, mentre l'Empoli ha distaccato il Palermo, in media arrivato terzultimo, di 8 punti, garantedosi la permanenza in Serie A. Infine, nella zona Europa League, ci sono Atalanta, Inter e Milan a distanza in media di un punto l'una dall'altra confermando come tutte queste squadre avvessero buone probabilità di accedere alla competezione europea, e che quella parte di classifica sia

quella soggetta a maggiore variabilità.

Domenica 28 Maggio si è disputata l'ultima giornata della Serie A 2016-17 e la classifica finale è riportata in Tabella 3.11. La sorpresa del finale di campionato è stata il Crotone che ha superato Palermo ed Empoli riuscendo ad evitare la retrocessione, evento per il quale avevamo stimato una probabilità molto bassa (3,6%), con i punti guadagnati battendo la Lazio proprio nell'ultima giornata. Il Crotone ha raccolto nelle ultime 8 giornate ben 17 punti, tanti quanti ne aveva raccolti nelle precedenti 30 partite, con una media di 2.125 punti a partita, quasi ugaule a quella che ha avuto nelle stesse giornate la Juventus (2.25). Si tratta chiaramente di un andamento anomalo e poco probabile, che conferma quindi che la stima della probabiltà della salvezza del Crotone doveva essere molto bassa. Vediamo inoltre che al suo posto è retrocesso proprio l'Empoli, che, come già osservato, nelle 82 simulazioni in cui una tra Crotone, Pescara e Palermo ha ottenuto la salvezza, è retorocesso in 81 casi. Per quanto riguarda la zona centrale della classfica non ci sono grandi differenze tra quella stimata e quella verificatasi realmente, spiccano solo il Sassuolo che ha ottenuto 6 punti in più rispetto a quelli stimati, e la Sampdoria che ne ha ottenuti invece 6 in meno, mentre nel resto dei casi abbiamo pochi punti di differenza. Salendo verso la cima della classifica, l'Inter non si è qualificata per l'Europa League, mentre si è qualificato il Milan che aveva una probabilità inferiore (64,1% contro 39,4%), ed entrambe hanno totalizzato meno punti di quanto stimato, rispettivamente 6 e 4 punti in meno. Infine, per quanto riguarda le prime tre posizioni, nonstante il Napoli abbia totalizzato 7 punti in più di quelli stimati, non è risucito a superare la Roma, confermando le posizioni stimate.

In conclusione, nonstante ci siano delle differenze (si veda Tabella 3.12), la classifica stimata rispecchia abbastanza bene quella finale, soprattutto per quanto riguarda i piazzamenti delle squadre nelle varie zone di interesse della classifca.

Squadra	Punti	Squadra	Punti
Juventus	91	Cagliari	47
Roma	87	Sassuolo	46
Napoli	86	Udinese	45
Atalanta	72	Chievo	43
Lazio	70	Bologna	41
Milan	63	Genoa	36
Inter	62	Crotone	34
Fiorentina	60	Empoli	32
Torino	53	Palermo	26
Sampdoria	48	Pescara	18

Tabella 3.11: La classifica finale della Serie A 2016-17.

Squadra	Differenza punti	Squadra	Differenza punti
Juventus	-1	Udinese	-2
Roma	3	Chievo	-3
Napoli	7	Cagliari	3
Lazio	-3	Bologna	-1
Atalanta	3	Sassuolo	6
Inter	-6	Genoa	0
Milan	-4	Empoli	1
Fiorentina	-4	Palermo	3
Sampdoria	-6	Crotone	12
Torino	1	Pescara	1

Tabella 3.12: Differenza di punti tra classifca finale reale e classifica finale stimata tramite Monte Carlo.

Conclusioni

Il modello statistico proposto presenta svariate applicazioni e ha il vantaggio di essere molto semplice. Per queste ragioni, nella letteratura riguardo la modellazione di risultati calcisitici, il modello Dixon-Coles è spesso preso in considerazione, e ne vengono proposte leggere modifiche o addattamenti, come l'inclusione dell'effetto di cartellini rossi (Stenerud, 2015).

Lo stesso Dixon con l'aiuto di Robinson, propone di modificare i parametri d'attacco nel corso della singola partita, perchè ha osservato come nel corso dell'incontro la frequenza dei gol segnati aumenta nel corso della partita (Dixon e Robinson, 1998) o nei minuti successivi alla realizzazione di una marcatura. Oppure ancora, per quanto spiegato nel Paragrafo 2.1.3, alcuni modelli vengono costruti sulla base della distribuzione Binomiale Negativa.

Ci sono poi ricerche per proporre indici diversi rispetto a (2.14) da usare nella valutazione delle probabilità stimate per gli esiti, come ad esempio il *ranked probability score* (Constantinuo e Fenton, 2012), che pesa in modo diverso gli errori di previsione sulla base della distanza dall'esito esatto.

Per quanto riguarda invece la valutazione e il confronto tra squadre, un indice proposto e molto usato, proveninete dal mondo degli scacchi, è l'indice ELO (Wikipedia, 2016).

Per concludere, inoltre, si possono trovare anche diversi nuovi modelli proposti basati su un approccio diverso, l'inferenza Bayesiana, che non ha una visione frequentistica delle probabilità ma più come livello di fiducia della realizzazione di un determinato evento (come ad esempio in Bååth, 2015).

Bibliografia

Azzalini, A. (2001). *Inferenza Statistica*. Milano: Springer Verlag.

Bååth, R. (2015). *Modeling match results in soccer using a hierarchical Bayesian Poisson model*. Technical Report. Lund University Cognitive Science.

Chiodi, M. (2000). *Tecniche di Simulazione Statistica*. Istituto di Statistica, Facoltà di Economia di Palermo.

Constantinuo, A.C. e N.E. Fenton (2012). «Solving the problem of inadequate scoring rules for assessing probabilistic football forecast models». In: *Journal of Quantitative Analysis in Sports*.

Curley, James (2016). *English and European Soccer Results 1871-2016*. R package version 0.1.5. URL: https://CRAN.R-project.org/package=engsoccerdata.

Dixon, M.J. e S.G. Coles (1997). «Modelling association football scores and inefficiencies in the football betting market». In: *Journal of the Royal Statical Society*, pp. 265–280.

Dixon, M.J. e M.E. Robinson (1998). «A birth process model for association football matches». In: *The Statistician*, pp. 523–538.

Goumas, C. (2015). «Modelling home advantage for individual teams in UEFA Champions League football». In: *Journal of Sport and Health Science*.

Karlis, D. e T. Ntzoufras (2000). «On modelling soccer data». In: *Student*, pp. 229–244.

Legaz-Arrese, A., D. Moliner-Urdiales e D. Munguía-Izquierdo (2013). «Home advantage and sports performance: evidence, causes and psychological implications». In: *Universitas Psychologica Panamerican Journal of Psychology*, pp. 933–943.

Maher, M.J. (1982). «Modelling association football scores». In: *Statistica Neerlandica*, pp. 109–118.

Moroney, M.J. (1951). *Facts from Figures*. Cap. 8 Goals, Floods, and Horsekicks - The Poisson Distribution, pp. 96–107.

Pace, L. e A. Salvan (2001). *Introduzione alla statistica II*. Milano: Cedam.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Rubinstein, R.Y. e D.K. Kroese (1981). *Simulation and the monte carlo method*.

Stenerud, S.G. (2015). *A study on soccer prediction using goals and shots on target*. Master of Science in Physics and Mathematics. Norwegian University of Science e Technology.

Wikipedia (2016). *World Football Elo Rating*. URL: https://it.wikipedia.org/wiki/World_Football_Elo_Ratings.

Appendice A

Codice R utilizzato

Codice A.1: Creazione di un dataset con i dati desiderati tramite le informazioni presenti nel paccheto `engsoccerdata`.

```
CreateDataFrame<- function(season,league,div=levels(league$division))
{

library(engsoccerdata)

if(league[1,1:8]==england[1,1:8])
{
for(j in 1:length(div))
{
if(j==1)
leag=league[league$division==div[1],]
else
leag=rbind(leag,league[(league$division==div[j]),])
}
}
else
{
leag=league
}
```

```
leag$home=as.character(leag$home)
leag$visitor=as.character(leag$visitor)



for(i in 1:length(season))
{
if(i==1)
table=leag[leag$Season==season[1],]
else
table=rbind(table,leag[(leag$Season==season[i]),])
}

table$home=as.factor(table$home)
table$visitor=as.factor(table$visitor)
teams=unique(c(levels(table$home),levels(table$visitor)))

HomeMatch=model.matrix(~ home-1 ,data=table)
AwayMatch=model.matrix(~ visitor-1 ,data=table)

HT=as.character(table$home)
AT=as.character(table$visitor)

date=as.Date(table$Date)

return(list(homeGoals=table$hgoal,awayGoals=table$vgoal,homeTeams=HT,
    awayTeams=AT,teams=teams, dummyHome=HomeMatch, dummyAway=AwayMatch,
    Date=date))
}
```

Codice A.2: Creazione di un dataset con i dati desiderati tramite i file csv scaricabili dal sito *football-data.co.uk/data.php*.

```
CreateDataFrame_footballdata.co.uk <- function()
{
table=read.csv(file.choose(),h=T)
HomeMatch = model.matrix(~ HomeTeam - 1, data=table)
AwayMatch = model.matrix(~ AwayTeam -1, data=table)
```

```
teams = unique(c(levels(table$HomeTeam), levels(table$AwayTeam)))
HT=as.character(table$HomeTeam)
AT=as.character(table$AwayTeam)
date=as.Date(table$Date, "%d/%m/%y")

return(list(homeGoals=table$FTHG,awayGoals=table$FTAG,homeTeams=HT,
    awayTeams=AT,teams=teams, dummyHome=HomeMatch, dummyAway=AwayMatch,
    Date=date))
}
```

Codice A.3: Codifica della funzione τ (2.7).

```
tau <- function(X, Y, lambda, mu, rho)
{
out <- rep(1,length(X))
out[((X == 0) & (Y == 0))] <- 1 - (lambda[((X == 0) & (Y == 0))]*mu[((X ==
     0) & (Y == 0))]*rho)
out[((X == 0) & (Y == 1))] <- 1 + (lambda[((X == 0) & (Y == 1))]*rho)
out[((X == 1) & (Y == 0))] <- 1 + (mu[((X == 1) & (Y == 0))]*rho)
out[((X == 1) & (Y == 1))] <- 1 - rho
return(out)
}
```

Codice A.4: Codifica della funzione $\phi(t)$ (2.12).

```
phi<-function(xi,data,ref.date)
{
t=(as.numeric(ref.date-data$Date))
weight=exp(-1*xi*t)
weight[t<=0]=0
return(weight)
}
```

Codice A.5: Codifica della funzione log verosimiglianza (2.6).

```
logLik_DC<- function(X, Y, lambda, mu, rho=0, phi=1)
{
```

```
sum(phi*(log(tau(X, Y, lambda, mu, rho)) + log(dpois(X, lambda)) + log(
    dpois(Y, mu))))
}
```

Codice A.6: Codifica della funzione di probabilita $p_{xy}(X = x, Y = y)$.

```
dDC<-function(lambda,mu,x,y,tau=1)
{
tau*(exp(-lambda)*lambda^x/factorial(x))*(exp(-mu)*mu^y/factorial(y))
}
```

Codice A.7: Codifica della funzioni utili per calcolare $S(\xi)$(2.14).

```
genTheta<-function(data,t)
{
k=length(data$homeGoals)
theta=rep(0,k)
if(t=="Home")
theta[data$homeGoals>data$awayGoals]=1
if(t=="Draw")
theta[data$homeGoals==data$awayGoals]=1
if(t=="Away")
theta[data$homeGoals<data$awayGoals]=1
return(theta)
}

genP<-function(data,mle)
{
k=length(data$homeGoals)
p=matrix(rep(1,(k*3)),nrow=k)
for(i in 1:k){
gen.p=p_MatchResult(mle,data$homeTeams[i],data$awayTeams[i])
p[i,1]=gen.p[1]
p[i,2]=gen.p[2]
p[i,3]=gen.p[3]
}
return(p)
}
```

```
S<-function(thetaHome,thetaDraw,thetaAway,p)
{
p.H=p[,1]
index=which(p.H==0)
p.H[index]=1
p.D=p[,2]
index=which(p.D==0)
p.D[index]=1
p.A=p[,3]
index=which(p.A==0)
p.A[index]=1
return(sum(thetaHome*log(p.H)+thetaDraw*log(p.D)+thetaAway*log(p.A)))
}
```

Codice A.8: Ottimizzazione del parametro ξ.

```
OptimizeXi<-function(data)
{
n=length(data$teams)
par=c(0.1,0.1,rep(0.2,n-1),rep(0.2,n))

tH=genTheta(data,"Home")
tD=genTheta(data,"Draw")
tA=genTheta(data,"Away")
t.k=unique(data$Date)

PL=rep(NA,100)
vec_xi=rep(NA,100)

z=1
s.val=NULL
xi=0
x=0

while(z<30)
{
```

```
cat("------xi:",x,"-------",fill=TRUE)
P.fin=matrix(rep(1,(length(data$homeGoals)*3)),ncol=3)
k=as.integer(length(t.k)/3)

while(k<(length(t.k)-10))
{
index=which(data$Date==t.k[k])
optim=nlminb(par,DC_logLik,data=data, phi=phi(x,data,ref.date=t.k[k]))

#genP usa funzione MatchGol che richede il passaggio di una lista con una
    tabella con i coefficenti e i nomi

coeff.atk=optim$par[3:(3+n-2)]
coeff.atk=c(coeff.atk,-sum(coeff.atk))
coeff.atk=coeff.atk+1
coeff.def=optim$par[(2+n):(2+n+n-1)]
coeff.def=coeff.def-1
mat=data.frame(teams=data$teams,Atk=coeff.atk,Def=coeff.def)
obj=list(table=mat,coeff.home=optim$par[1],rho=optim$par[2])


P.fin[index:which(data$Date==t.k[(k+10)]),]=genP(data,obj)[index:which(
    data$Date==t.k[(k+10)]),]
k=k+10

}

index=which(data$Date>=t.k[k])
optim=nlminb(par,DC_logLik,data=data, phi=phi(x,data,ref.date=t.k[k]))


coeff.atk=optim$par[3:(3+n-2)]
coeff.atk=c(coeff.atk,-sum(coeff.atk))
coeff.atk=coeff.atk+1
coeff.def=optim$par[(2+n):(2+n+n-1)]
```

```
coeff.def=coeff.def-1
mat=data.frame(teams=data$teams,Atk=coeff.atk,Def=coeff.def)
obj=list(table=mat,coeff.home=optim$par[1],rho=optim$par[2])



P.fin[index,]=genP(data,obj)[index,]

index=which(data$Date>t.k[as.integer(length(t.k)/2)])
s.val2=S(tH[index],tD[index],tA[index],P.fin[index,])

cat(s.val2,fill=TRUE)

if((is.null(s.val))||(s.val2>s.val))
{
xi=x
s.val=s.val2
}

PL[z]=s.val2
vec_xi[z]=x

z=z+1
x=x+0.0002
plot(vec_xi,PL,type="l")

}

return(list(optim.xi=xi,vec.xi=vec_xi,PLL=PL))
}
```

Per quanto riguarda questa funzione, la prima data considerata da cui iniziare le stime e la distanza tra una data e l'altra sono da aggiustare sulla base della grandezza del dataset considerato. Nel Capitolo 3 sono presenti delle considerazioni che aiutano a stabilire la data di partenza, ma per quanto riguarda il salto da una data alle succesive, si deve cercare un compromomesso tra non perdere troppi incontri e velocità computazionale.

Codice A.9: Calcolo delle stime di massima verosimiglianza α_i, β_i, γ.

```
DC_logLik<- function(par,data,phi=1)
{
coef.home = par[1]
rho = par[2]
n=length(data$teams)
coef.attack = par[3:(n+1)]
coef.attack = c(coef.attack,-sum(coef.attack))
coef.defence =par[(n+2):length(par)]

lambda= as.vector(exp(data$dummyHome %*% coef.attack +  data$dummyAway %*%
     coef.defence + coef.home))
mu=as.vector(exp(data$dummyAway %*% coef.attack + data$dummyHome %*% coef.
    defence))

if ((rho > max(c(-1/lambda,-1/mu))) & (rho < min(c(1,1/(lambda*mu)))))
return(logLik_DC(data$homeGoals,data$awayGoals, lambda, mu, rho,phi)*-1)
else
return(Inf)
}



DC_MaxLikEst<- function(logLik,data,xi=0,ref.date=max(data$Date))
{
n=length(data$teams)
if(!require(numDeriv)) {install.packages("numDeriv")
library(numDeriv)}
else{library(numDeriv)}

par=c(0.1,0.1,rep(0.2,n-1),rep(0.2,n))

optim=nlminb(par,logLik,data=data,phi=phi(xi,data,ref.date))
hess=hessian(func=logLik,x=par,data=data,phi=phi(xi,data,ref.date))

coeff.home=optim$par[1]
coeff.rho=optim$par[2]
```

```
coeff.atk=optim$par[3:(3+n-2)]
coeff.atk=c(coeff.atk,-sum(coeff.atk))
coeff.atk=coeff.atk+1

coeff.def=optim$par[(2+n):(2+n+n-1)]
coeff.def=coeff.def-1

if(xi==0)
{
j=solve(hess)
st.er_h=sqrt(j[1,1])
st.er_r=sqrt(j[2,2])
st.er_a=sqrt(diag(j))[3:(n+3-2)]
st.er_a=c(st.er_a,sqrt(sum(j[3:(3+n-2),(3:3+n-2)])))
st.er_d=sqrt(diag(j))[(n+2):(2+n+n-1)]

mat=data.frame(teams=data$teams,Atk=coeff.atk,Std.Er_Atk=st.er_a,Def=coeff
    .def,Std.Er_Def=st.er_d)
}
else
mat=data.frame(teams=data$teams,Atk=coeff.atk,Def=coeff.def)

cat("Home Advantage Coefficent (Gamma)",coeff.home,sep=": ",fill = TRUE)
cat("Correlation Parameter (Rho)",coeff.rho, sep=": ",fill=TRUE)
cat("Xi",xi,sep=": ",fill=TRUE)

print(mat)

return(list(coeff.atk=coeff.atk,coeff.def=coeff.def,coeff.home=coeff.home,
    rho=coeff.rho,loglik.value=optim$value,table=mat,hessian=hess))
}
```

Il vincolo alternativo (2.9), risulta utile in quanto ci consente di usare la funzione di ottimizzazione presente in R `nlminb`. Con l'altro vincolo avremmo dovuto usare la funzione `auglag` contenuta nel pacchetto `alabama`, che

aumentava di molto però i tempi necessari a calcolare le stime.

Codice A.10: Serie storica delle stime di α_i, β_i, γ.

```
dinamicMle<-function(data,xi)
{
n=length(data$teams)
dates=unique(data$Date)
k=length(dates)


ref=as.integer(k/4)

mle=DC_MaxLikEst(DC_logLik,data,xi,ref.date = dates[ref])
atkmat=mle$coeff.atk
defmat=mle$coeff.def
gamma=mle$coeff.home
vecdate=dates[ref]
ref=ref+10

while(ref<=k-10)
{
mle=DC_MaxLikEst(DC_logLik,data,xi,ref.date = dates[ref])
atkmat=cbind(atkmat, mle$coeff.atk)
defmat=cbind(defmat, mle$coeff.def)
gamma=c(gamma,mle$coeff.home)
vecdate=c(vecdate,dates[ref])
ref=ref+10
}

ref=k
mle=DC_MaxLikEst(DC_logLik,data,xi,ref.date = dates[ref])
atkmat=cbind(atkmat, mle$coeff.atk)
defmat=cbind(defmat, mle$coeff.def)
gamma=c(gamma,mle$coeff.home)
vecdate=c(vecdate,dates[ref])
```

```
return(list(atkmat=atkmat,defmat=defmat,home=gamma,t=vecdate))
}
```

Codice A.11: Stima della probabilità del risultato $x - y$.

```
MatchGol<-function(mle,teamHome,teamAway,maxgol=4)
\{
a=NA
h=NA
name=mle\$table\$teams
a=which(name==teamAway)
h=which(name==teamHome)
lambda=exp(mle\$table\$Atk[h]+mle\$table\$Def[a]+mle\$coeff.home)
mu=exp(mle\$table\$Atk[a]+mle\$table\$Def[h])
p=rep(NA,maxgol*maxgol)
gh=rep(NA,maxgol*maxgol)
ga=rep(NA,maxgol*maxgol)
z=1
for(i in 0:maxgol)
\{
for(r in 0:maxgol)
gh[z+r]=i
for(j in 0:maxgol)\{
ga[z]=j
p[z]=dDC(lambda,mu,i,j,tau(i,j,lambda,mu,mle\$rho))
z=z+1 \}
\}
res=data.frame(golHome=gh,GolAway=ga,p=p)
best\_index=which(res\$p==max(res\$p))
best=res[best\_index,1:3]
return(list(table=res,lambda=lambda,mu=mu,best=best))
\}


p_singleResult<-function(mle,teamHome,teamAway,golHome,golAway)
{
h=which(mle$table$teams==teamHome)
```

```
a=which(mle$table$teams==teamAway)
lambda=exp(mle$table$Atk[h]+mle$table$Def[a]+mle$coeff.home)
mu=exp(mle$table$Atk[a]+mle$table$Def[h])
return(dDC(lambda,mu,golHome,golAway,tau(golHome,golAway,lambda,mu,mle$rho
    )))
}
```

Codice A.12: Stima delle probabilità degli esiti V-P-S.

```
p_MatchResult<- function(mle,teamHome,teamAway)
{
res=MatchGol(mle,teamHome,teamAway,maxgol=10)
r=rep(0,3)
n=length(r)
w=res$table$golHome>res$table$GolAway
l=res$table$golHome<res$table$GolAway
d=res$table$golHome==res$table$GolAway
score=c(sum(w*res$table$p),sum(d*res$table$p),sum(l*res$table$p))
names(score)=c("Vittoria","Pareggio","Sconfitta")
return(score)
}
```

Codice A.13: Simulazione di un risultato per le partite indicate.

```
Simulate<-function(mle,listHome,listAway)
{
sim=integer(length(listHome))
for(i in 1:length(listHome))
{
teamH=listHome[i]
teamA=listAway[i]
sim[i]=rOut(1,mle,teamH,teamA)
}

res=MatchGol(mle,listHome[1],listAway[1],maxgol = 10)
golH=res$table$golHome
golA=res$table$GolAway
simulation=cbind(golH[sim],golA[sim])
```

```
esito=character(length(listHome))
esito[simulation[,1]>simulation[,2]]="W"
esito[simulation[,1]==simulation[,2]]="D"
esito[simulation[,1]<simulation[,2]]="L"

data.frame(Home=listHome,Visitor=listAway,hgol=simulation[,1],vgol=
    simulation[,2],outcome=esito)
}
```

Codice A.14: Creazione di una classifica ordinata.

```
points<-function(esito)
{
teams=unique(c(levels(esito$Home),levels(esito$Visitor)))
points=data.frame(teams=teams,points=rep(0,length(teams)))
for(i in 1:length(esito$Home))
{
if(esito$outcome[i]=="W")
{
team=esito$Home[i]
points$points[points$teams==team]=points$points[points$teams==team]+3
}

if(esito$outcome[i]=="L")
{
team=esito$Visitor[i]
points$points[points$teams==team]=points$points[points$teams==team]+3
}

if(esito$outcome[i]=="D")
{
team=esito$Home[i]
points$points[points$teams==team]=points$points[points$teams==team]+1
team=esito$Visitor[i]
points$points[points$teams==team]=points$points[points$teams==team]+1
}
```

```
}

return(points)
}

MakeRanking<-function(data)
{
listHome=as.factor(dataset$awayTeams)
listAway=as.factor(dataset$awayTeams)
teams=unique(c(levels(listHome),levels(listAway)))
points=data.frame(teams=teams,points=rep(0,length(teams)))

for(i in 1:length(dataset$homeGoals))
{
if(dataset$homeGoals[i]>dataset$awayGoals[i])
{
team=dataset$homeTeams[i]
points$points[points$teams==team]=points$points[points$teams==team]+3
}

if(dataset$homeGoals[i]<dataset$awayGoals[i])
{
team=dataset$awayTeams[i]
points$points[points$teams==team]=points$points[points$teams==team]+3
}

if(dataset$homeGoals[i]==dataset$awayGoals[i])
{
team=dataset$homeTeams[i]
points$points[points$teams==team]=points$points[points$teams==team]+1
team=dataset$awayTeams[i]
points$points[points$teams==team]=points$points[points$teams==team]+1
}
}
return(points)
}
```

```
orderTable<-function(table)
{
return(table[order(table$points,decreasing = TRUE),])
}
```

Codice A.15: Generazione casuale di risultati.

```
rOut<-function(n,mle,teamHome,teamAway)
{
p=MatchGol(mle,teamHome,teamAway,maxgol = 10)$table$p
sample(c(1:121),size=n,replace = TRUE, p)
}

Simulate<-function(mle,listHome,listAway)
{
sim=integer(length(listHome))
for(i in 1:length(listHome))
{
teamH=listHome[i]
teamA=listAway[i]
sim[i]=rOut(1,mle,teamH,teamA)
}

res=MatchGol(mle,listHome[1],listAway[1],maxgol = 10)
golH=res$table$golHome
golA=res$table$GolAway
simulation=cbind(golH[sim],golA[sim])

esito=character(length(listHome))
esito[simulation[,1]>simulation[,2]]="W"
esito[simulation[,1]==simulation[,2]]="D"
esito[simulation[,1]<simulation[,2]]="L"

data.frame(Home=listHome,Visitor=listAway,hgol=simulation[,1],vgol=
    simulation[,2],outcome=esito)
```

```
}



points<-function(esito)
{
teams=unique(c(levels(esito$Home),levels(esito$Visitor)))
points=data.frame(teams=teams,points=rep(0,length(teams)))
for(i in 1:length(esito$Home))
{
if(esito$outcome[i]=="W")
{
team=esito$Home[i]
points$points[points$teams==team]=points$points[points$teams==team]+3
}

if(esito$outcome[i]=="L")
{
team=esito$Visitor[i]
points$points[points$teams==team]=points$points[points$teams==team]+3
}

if(esito$outcome[i]=="D")
{
team=esito$Home[i]
points$points[points$teams==team]=points$points[points$teams==team]+1
team=esito$Visitor[i]
points$points[points$teams==team]=points$points[points$teams==team]+1
}
}

return(points)
}
```

Codice A.16: Simulazione Monte Carlo.

```
MonteCarlo_Simulation<-function(n,mle,listHome,listAway,champ.place=3,
    eurolg.place=3,down.place=3,partial=FALSE,ranking)
```

```
{
m=nlevels(unique(as.factor(listHome),as.factor(listAway)))
win=rep(0,m)
champ=rep(0,m)
eurolg=rep(0,m)
down=rep(0,m)

sim=points(Simulate(mle,listHome,listAway))
if(partial){
sim[,2]=ranking[,2]+sim[,2]
}
ordered=orderTable(sim)

win[as.character(mle$table$teams) %in% as.character(ordered$teams[1])]=win
    [as.character(mle$table$teams) %in% as.character(ordered$teams[1])]+1
champ[which(mle$table$teams %in% ordered$teams[1:champ.place])]=  champ[
    mle$table$teams %in% ordered$teams[1:champ.place]]+1
eurolg[mle$table$teams %in% ordered$teams[(champ.place+1):(champ.place+
    eurolg.place)]]=  eurolg[mle$table$teams %in% ordered$teams[(champ.
    place+1):(champ.place+eurolg.place)]]+1
down[mle$table$teams %in% ordered$teams[(m-down.place+1):m]]=  down[mle$
    table$teams %in% ordered$teams[(m-down.place+1):m]]+1

mat=sim$points

for(i in 2:n)
{

sim=points(Simulate(mle,listHome,listAway))
if(partial){
sim[,2]=ranking[,2]+sim[,2]
}
ordered=orderTable(sim)
win[mle$table$teams %in% ordered$teams[1]]=win[mle$table$teams %in%
    ordered$teams[1]]+1
champ[which(mle$table$teams %in% ordered$teams[1:champ.place])]=  champ[
```

```
    mle$table$teams %in% ordered$teams[1:champ.place]]+1
eurolg[mle$table$teams %in% ordered$teams[(champ.place+1):(champ.place+
    eurolg.place)]]=  eurolg[mle$table$teams %in% ordered$teams[(champ.
    place+1):(champ.place+eurolg.place)]]+1
down[mle$table$teams %in% ordered$teams[(m-down.place+1):m]]=  down[mle$
    table$teams %in% ordered$teams[(m-down.place+1):m]]+1

mat=rbind(mat,sim$points)

}
expected=as.integer(apply(mat,2,mean))
final=data.frame(teams=mle$table$teams,points=expected)
placement=data.frame(teams=mle$table$teams,win,champ,eurolg,down)
return(list(orderTable(final),placement))
}
```