Sección de Matemáticas
Universidad de La Laguna

Juan Diego Fernández García

Modelos probabilísticos para la estimación de resultados deportivos

Probabilistic models for the estimation of sport score

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Julio de 2019

Dirigido por
Carlos M. González Alcón

Carlos M. González Alcón
Estadística e investigación operativa
Universidad de La Laguna
38200 La Laguna, Tenerife

Agradecimientos

A todos los profesores y compañeros de los que he podido aprender cosas a lo largo del grado.

Juan Diego Fernández García
La Laguna, July 10, 2019

Resumen · Abstract

Resumen

En este trabajo se presentarán dos modelos probabilísticos cuya principal herramienta es la distribución de Poisson. El objetivo de estas dos propuestas es asignar a cada partido de fútbol una distribución de probabilidad para los posibles resultados. En la primera propuesta asumiremos independencia entre los goles que marcarán los equipos enfrentados. El modelo alternativo está fundamentado en la Poisson Bivariante, que asume correlación entre el número de goles de ambos equipos. Se analizan y comparan ambos métodos utilizando los datos de LaLiga y La Liga Iberdrola. Propondremos también un modelo de partido como cadena de Markov que nos permita hacer simulaciones. En particular en la presente memoria, constarán ejemplos con datos de LaLiga y La Liga Iberdrola 2018-19.

Palabras clave: *Poisson univariada – Poisson bivariada – Regresión – Lenguaje de programación R – Cadena de Markov – Matriz estocástica – Probabilidad de transición*

Abstract

This work will present two probabilistic models whose main tool is the distribution of Poisson. The aim of these two proposals is to assign each football match a probability distribution for the possible outcomes. In the first proposal we will assume independence among the goals that will mark the opposing teams. The alternative model is based on the bivariate Poisson, which assumes correlation between the number of goals of both teams. Both methods are analyzed and compared using data from LaLiga and La Liga Iberdrola. We will also propose a party model as a Markov chain that allows us to do simulations. In particular in this report, examples with data from LaLiga and La Liga Iberdrola 2018-19 will be included.

Keywords: *Univariate Poisson – bivariate Poisson – Regression – Programming Language R – Markov Chain – Stochastic Matrix – Transition Probability*

Contenido

Introducción

Si queremos considerar algún deporte como universal, este es el fútbol. Nuestro primer objetivo es desarrollar un modelo probabilístico con el fin de predecir resultados en deportes en los que se enfrentan dos partes. Pueden ser individuales o por equipos, donde la victoria se decide mediante tanteo.

Hemos investigado acerca de posibles modelos que estimen la probabilidad de un partido de fútbol, nos encontramos artículos donde la herramienta principal es la distribución de Poisson, en la mayoría de casos univariante. En los primeros meses de iniciar este proyecto, nos cuestionábamos si en general había correlación entre los goles que marcan uno y otro equipo , para poder comprobar este punto necesitaríamos datos de bastantes enfrentamientos entre dos mismos equipos temporalmente no demasiado lejanos, cosa que no suele ser posible. De esta forma tomamos la decisión de plantear dos modelos, en uno considerando independencia y otro considerando correlación, con ambos haremos el mismo trabajo de la forma que veremos en esta memoria y finalmente veremos si hay diferencias significativas en los dos modelos aplicados.

En cuanto a modelos bivariantes, son muy útiles cuando se asume que existe correlación entre las dos variables. Lamentablemente la literatura sobre tales modelos es escasa debido a problemas computacionales complicados en su puesta en práctica. Los modelos en los que se usa la Poisson bivariante pueden ser ampliados para tener covariables en cuenta, se utiliza la regresión para estimar los parámetros necesarios de la Poisson bivariante.

En la parte final del trabajo haremos uso de cadenas de Markov para simular el desarrollo y resultado de un partido, cuyos parámetros de entrada pueden ser obtenidos a partir de los diferentes enfoques presentados anteriormente.

Los modelos desarrollados se han probado con los datos proporcionados por la primera vuelta de LaLiga y La Liga Iberdrola. Los métodos se han implementado con ayuda del lenguaje de programación R, así como la elaboración de ejemplos y gráficas.

1

Modelos basados en la distribución de Poisson

1.1 Marco teórico

1.1.1 La poisson univariada

Sea X una variable aleatoria discreta, esta sigue una distribución de Poisson de parámetro λ cuando su función de probabilidad viene dada por:

$$f(x) = P(X = x) = \frac{e^{-\lambda}\lambda^x}{x!}, \quad \text{con} \quad x = 0, 1, 2, 3 \dots$$

Una posible interpretación de la distribución de Poisson es tomarla como una distribución de probabilidad discreta que expresa la probabilidad de que suceda un determinado número de eventos durante cierto periodo de tiempo.

Proposición 1.1. *La media y la varianza son iguales a su parámetro,* $E(X) = \lambda = Var(X)$.

Proposición 1.2. *Las probabilidades aumentan hasta el mayor entero menor que* λ*, a partir de entonces, disminuyen.*

1.1.2 La poisson bidimensional

Sean X_k, tal que $k = 1, 2, 3$. Estas son variables aleatorias independientes, cada una de ellas tiene asignado un parámetro correspondiente λ_k positivo.

Se consideran ahora dos nuevas variables aleatorias, suma cada una de dos de las anteriores $X = X_1 + X_3$, $Y = X_2 + X_3$.

Proposición 1.3. *Decimos que* (X, Y) *sigue una distribución de Poisson bidimensional* $BP(\lambda_1, \lambda_2, \lambda_3)$ *si su función de probabilidad es la siguiente:*

$$f_{BP}(x,y) = P(X=x, Y=y) = e^{-(\lambda_1+\lambda_2+\lambda_3)} \frac{\lambda_1^x}{x!} \frac{\lambda_2^y}{y!} \sum_{k=0}^{\min(x,y)} \binom{x}{k} \binom{y}{k} k! \left(\frac{\lambda_3}{\lambda_1\lambda_2}\right)^k$$

$$x, y = 0, 1, 2 \ldots$$

Por separado, cada una de las variables aleatorias sigue una distribución de Poisson unidimensional de parámetros $\lambda_1 + \lambda_3$ y $\lambda_2 + \lambda_3$.

Proposición 1.4. *Dada una Poisson bivariante* (X, Y) *de parámetros* $\lambda_1, \lambda_2, \lambda_3$ *tenemos que:*

$$E(X) = \lambda_1 + \lambda_3$$
$$E(Y) = \lambda_2 + \lambda_3$$
$$Cov(X, Y) = \lambda_3.$$

Como consecuencia de esta última, λ_3 es una medida de dependencia entre X e Y.

Nos encontramos ante dos escenarios posibles:

En primer lugar el caso en el que $\lambda_3 = 0$, por consiguiente X e Y son independientes, de tal forma que este caso queda reducido a dos distribuciones de Poisson univariantes.

La alternativa es que estemos ante $\lambda_3 > 0$, donde debemos aplicar un modelo de Poisson Bivariado.

Estudiaremos primero el caso $\lambda_3 = 0$.

1.1.3 Modelo de Regresión de Poisson

El objetivo es modelizar una cantidad discreta usando la distribución de Poisson para estudiar si ciertas variables explicativas influyen en la variable respuesta y cómo lo hacen.

La variable discreta en este caso es el número de goles de un partido de fútbol. Entonces consideraremos nuestra variable respuesta NG=Número de

goles= $0, 1, 2...$ y estudiaremos su relación con VE=Variables explicativas. Planteamos el siguiente modelo:

$$\lambda(x) = E(NG|VE = x) \quad x \in VE,$$

esto es, la media de goles condicionado a cierta variable explicativa x.

Se tiene que $NG \geq 0$, luego no procede usar un modelo lineal directo, debemos usar una función como enlace con dominio $(0, \infty)$, de modo que usaremos la función logaritmo. Tendríamos que:

$$g(\lambda(x, \beta)) = x'\beta,$$

donde $x'\beta$ denota el producto escalar del vector de variables explicativas y el vector de parámetros. Para que exista un intercepto consideraremos una primera variable explicativa con valor 1. Se toma $g(r) = \log(r)$ con $r \in (0, \infty)$ para que la función de regresión final sea

$$\lambda(x, \beta) = e^{x'\beta}.$$

Los parámetros de este modelo son los siguientes:

Exponencial del intercepto: Valor esperado de la respuesta cuando las variables explicativas numéricas valen 0.

Exponenciales de los coeficientes de las variables: tasas de incremento de la respuesta esperada en la variable (si es cuantitativa) o en la categoría (si es cualitativa).

El siguiente resultado afirma que si una componente de una variable explicativa aumenta n unidades, entonces la media en la variable de Poisson se eleva a n.

Proposición 1.5.

$$\frac{\lambda((x_1, \ldots, x_j + n, \ldots, x_p), \beta)}{\lambda((x_1, \ldots, x_j, \ldots, x_p), \beta)} = \frac{\exp(\beta_0 + x_1\beta_1 + \cdots + (x_j + n)\beta_j + \cdots + x_p\beta_p)}{\exp(\beta_0 + x_1\beta_1 + \cdots + x_j\beta_j + \cdots + x_p\beta_p)} = e^{n\beta_j}.$$

Proposición 1.6. *Sea* $(VE_1, NG_1), (VE_2, NG_2), \ldots, (VE_n, NG_n)$ *una muestra aleatoria simple de* (VE, NG)*, entonces* $NG_i \sim Poisson(\lambda(VE_i, \beta))$ *donde* $\lambda(x, \beta) = e^{x'\beta}$.

Estimación de los parámetros del modelo

Usaremos la función de máxima verosimilitud, que en términos de β tiene la siguiente forma:

$$L(\beta) = \prod_{i=1}^{n} e^{-\lambda(x_i,\beta)} \frac{\lambda(x_i,\beta)^{y_i}}{y_i!},$$

cuyo logaritmo neperiano es:

$$ln(L(\beta)) = \sum_{i=1}^{n} y_i x_i^{'} \beta - e^{x_i^{'}\beta}.$$

Si derivamos la función respecto a β e igualamos a 0 obtenemos las ecuaciones de verosimilitud:

$$\frac{\partial(\beta)}{\partial\beta} = -\sum_{i=1}^{n} x_i x_i^{'} \lambda(x_i,\beta).$$

Los estimadores de máxima verosimilitud son asintóticamente normales y centrados y su matriz de varianzas-covarianzas es precisamente la matriz hessiana cambiada de signo, esto es, la inversa de la matriz de información.

Como consecuencia podremos hacer inferencia sobre los parámetros del modelo. Sin embargo, no usaremos este modelo, sino el de la Poisson bidimensional, concretramente el desarrollado por Karlis y Ntzoufras (2003). Donde asumiremos dependencia entre las variables.

1.1.4 Regresión de Poisson bidimensional

Vamos a considerar un caso en general, donde que w_{ki} representa el valor de las variables explicativas usada en el modelo λ_{ki}. Mientras que β_k $(k = 1, 2, 3)$ representa el vector de coeficientes de regresión.

Utilizaremos la notación (GL, GV) para la variable respuesta, así como W para la variable explicativa. La adaptación al modelo a la forma bivariante es la siguiente para la obsevación i-ésima.

$$(GL_i, GV_i) \sim BP(\lambda_{1i}, \lambda_{2i}, \lambda_{3i}),$$

donde

$$\log(\lambda_{1i}) = w_{1i}'\beta_1$$
$$\log(\lambda_{2i}) = w_{2i}'\beta_2$$

$$\log(\lambda_{3i}) = w'_{3i}\beta_3.$$

Usaremos modelos con λ_3 constante, esto es, sin covariables sobre λ_3 para una interpretación más sencilla de los mismos.

Este modelo es iterativo, el método que utilizaremos para cada uno de los pasos i será el algoritmo EM que introduciremos a continuación.

Algoritmo EM

Las siglas que dan nombre a este algoritmo significan "Esperanza" y "Maximizar". Este método generalmente se utiliza para estimar valores ausentes en problemas de análisis multivariante. En resumen consiste en aumentar la muestra (datos observados) con algunos datos no observados para maximizar la verosimilitud de manera más sencilla. El algoritmo consta de dos pasos llamados "Paso E" y "Paso M".

Paso E: Sea un estimador inicial de los parámetros, calculamos la esperanza de las funciones de los valores ausentes que sin embargo sí estan presentes en la verosimilitud completa o aumentada. Dicha esperanza se calcula con respecto a la distribución de los valores ausentes dados en los valores observados y también respecto a las estimaciones iniciales. Cuando la verosimilitud completa es una función lineal de los valores ausentes, entonces se sustituyen estos por las esperanzas condicionadas a los valores observados y a los parámetros estimados.

Paso M: Dada la función lineal obtenida en el proceso anterior, ahora el objetivo es maximizar la verosimilitud. Veremos esto de manera más clara en la aplicación del modelo.

Con el valor obtenido en el paso M terminaríamos una iteración. Para la siguiente volveríamos al paso E y repetimos el proceso hasta tener una diferencia considerablemente pequeña.

1.1.5 Paquete estadístico de R: bivpois y la función lm.bp

Para implementar en R el modelo de regresión bivariante junto con el algoritmo EM vistos anteriormente, los propios creadores, Karlis y Ntzoufras, crearon un paquete en dicho lenguaje de programación. El nombre del paquete es bivpois y puede obtenerse en la dirección. http://www2.stat-athens.aueb.gr/~jbn/papers/paper14.htm. Además del paquete, los fundamentos teóricos se pueden encontrar en el artículo de los mismos autores. "Bivariate Poisson and Diagonal

Inflated Bivariate Poisson Regression Models in R" [2].

La función que utilizaremos es lm.bp , que tiene los siguientes argumentos:
lm.bp(l1, l2, l1l2=NULL, l3= 1, data, common.intercept=
FALSE, zeroL3=FALSE, maxit=300, pres=1e-8,
verbose=getOption("verbose"))

l1=fórmula de la forma $x \sim X_1 + \cdots + X_p$ para los parámetros de $\log(\lambda_1)$.

l2=fórmula de la forma $y \sim X_1 + \cdots + X_p$ para los parámetros de $\log(\lambda_2)$.

l1l2=fórmula de la forma $\sim X_1 + \cdots + X_p$ para los parámetros comunes de $\log(\lambda_1)$ y $\log(\lambda_2)$. Si $l1$ o $l2$ contiene a la variable explicativa entonces el modelo ajusta la interacción entre los parámetros.

l3=fórmula de la forma $\sim X_1 + \cdots + X_p$ para los parámetros de $\log(\lambda_3)$.

data: fichero de datos que almacena las variables del modelo.

common.intercept: es una función de decisión que devuelve *TRUE* si hay intercepto entre λ_1 y λ_2 y *FALSE* (que es su valor por defecto) en otro caso.

zeroL3: función de decisión, dictamina si $\lambda_3 = 0$. Por defecto tiene el valor *FALSE*

maxit: representa el número máximo de iteraciones, que por defecto es 300.

pres: precisión usada por el algoritmo EM Así la regla de parada será cuando la verosimilitud relativa sea menor que esta precisión.

verbose: es un argumento lógico, *TRUE* representa que los β_i deben calcularse mientras se ejecuta el EM. *FALSE* (por defecto) en caso contario. En este último caso solo se llevan a cabo el número de iteraciones el logaritmo de verosimilitud y la diferencia relativa de iteraciones previas.

El objetivo de la función es dar un objeto lista con información sobre el ajuste del modelo de regresión bivariante enunciado en secciones anteriores:

$$(GL_i, GV_i) \sim BP(\lambda_{1i}, \lambda_{2i}, \lambda_{3i}),$$

donde

$$\log(\lambda_{1i}) = w'_{1i}\beta_1$$
$$\log(\lambda_{2i}) = w'_{2i}\beta_2$$

$$\log(\lambda_{3i}) = w'_{3i}\beta_3.$$

Siendo $i = 1, \ldots, n$ con n el tamaño de la muestra. $\lambda_k = (\lambda_{k1}, \lambda_{k2}, \ldots, \lambda_{kn})$ con $k = 1, 2, 3$ son vectores de longitud n con la λ estimada en cada observación. w_1 y w_2 son matrices $n \times p$ con la información de las variables explicativas de λ_1 y λ_2. Análogamente w_3 es otra matriz que lleva la información de λ_3, en este caso de dimensión $n \times p$. Finalmente, los β_k $(k = 1, 2, 3)$ son los vectores de los parámetros usados para predecir los λ_k correspondientes.

Valores obtenidos

A continuación listaremos las componentes resultantes de aplicar la función *lm.bp*:

coefficients: Estimación de los parámetros para λ_k $k = 1, 2, 3$. Si se usa un factor, se obtienen estimadores según la codificación empleada.

fitted.values: Matriz $n \times 2$ con los valores ajustados para x e y siendo n el número de observaciones, esto es $\lambda_1 + \lambda_3$ y $\lambda_2 + \lambda_3$ respectivamente.

residuals: Matriz $n \times 2$ con los residuos del modelo para x e y. En este caso vienen dados por $x - E(x)$ e $y - E(y)$, recordando que $E(x) = \lambda_1 + \lambda_3$ y $E(y) = \lambda_2 + \lambda_3$.

$beta_k$ $(k = 1, 2, 3)$: Vectores que contienen los coeficientes implicados en predecir linealmente λ_k $k = 1, 2, 3$. Si *zeroL3=TRUE* el β_3 no se calcula.

λ_k $k = 1, 2, 3$: Vectores que contienen la estimación para cada k. Si *zeroL3=TRUE* el λ_3 toma el valor 0 y por tanto no es necesario mostrarlo.

loglikelihood: Vector que muesta la evolución del logaritmo de verosimilitud en cada paso del algoritmo EM

parameters: cantidad de parámetros.

iterations: cantidad de iteraciones.

1.2 Modelos propuestos

En esta sección propondremos un modelo sencillo que sólo utiliza distribuciones de Poisson en una variable suponiendo independencia entre las mismas. Estas

son los goles de cada uno de los equipos enfrentados. Luego desarrollaremos otro modelo algo más complejo usando la Poisson Bidimensional asumiendo que hay correlación positiva.

1.2.1 Modelo de Poisson en una variable

La distribución de Poisson permite calcular la probabilidad de cuántas veces ocurrirá un suceso durante un tiempo determinado. En nuestro caso el objetivo es que pronostique el número de goles en un partido completo. A partir de aquí, con este modelo podremos calcular probabilidades para un partido de la temporada siguiente a la que hemos tomado los datos. Usaremos el modelo del *Quinigol*, que es un tipo de apuesta deportiva. En el boleto hay 6 partidos de primera división y el objetivo es acertar el máximo número de resultados posibles. Para cada resultado se puede elegir los valores 0, 1, 2 o M, significando este último más de tres goles.

Como estamos utilizando la distribución de Poisson en una variable asumimos independencia entre las variables goles local y goles visitante. Por tanto la resolución que proponemos consiste en dar una Poisson para cada uno de los 20 equipos, cuyo parámetro λ se traduce como la media de goles de dicho equipo la temporada anterior. La probabilidad de que un partido quede con el resultado $i - j$ con $i, j = 0, 1, 2, M$. Será $P(\text{goles local=i}) \cdot P(\text{goles visitante=j})$.

Como datos usaremos los resultados de los partidos de la primera vuelta de la primera división española, esto es la mitad de una liga. Con esta información podemos ver cuál es la media de goles de cada equipo en la primera vuelta, dato que utilizaremos como parámetro de la distribución de Poisson que emplearemos para cada equipo. Esto es $\lambda_i = \frac{GF_i}{19}$. Queda la siguiente tabla en la que los equipos están ordenados por posición se lee de arriba a abajo cada una de las columnas empezando por la izquierda:

Con esto podemos ver cómo se distribuye la probabilidad de los posibles resultados de los partidos de la jornada 20, que es la primera de la segunda vuelta. Para las demás jornadas tomaremos la media de los goles de los 19 partidos anteriores, lo correspondiente a una vuelta de LaLiga en este caso. A medida que avance la temporada tendremos una muestra mayor, por ejemplo en la jornada 35 tendremos la media de goles de cada equipo tomada en una muestra de 34 partidos.

Equipo	λ	Equipo	λ
Barcelona	$2,789$	Valencia	$0,895$
At. Madrid	$1,421$	Levante	$1,579$
Sevilla	$1,632$	Athletic	$1,105$
R. Madrid	$1,474$	Valladolid	$0,895$
Alavés	$1,158$	Leganés	$0,895$
Getafe	$1,106$	Eibar	$1,105$
Betis	$1,157$	Celta	$1,632$
R. Sociedad	$1,211$	Rayo	$1,158$
Girona	$1,105$	Villareal	$1,053$
Espanyol	$1,105$	Huesca	$0,895$

Tabla 1.1. Media de goles por equipos de la primera vuelta de LaLiga 18-19

Ejemplo

Ejemplo 1.7. Calcular la probabilidad del partido de la jornada 20 entre el Real Madrid y el Sevilla basándonos en el modelo introductorio para la Poisson Univariante para eñ resultados 2-0, (ayudándonos de R), aclarar que existe una función en R llamada *dpois* que hace esta misma tarea, sin embargo en este ejemplo queremos verlo con mayor detenimiento:

```
x=2
lambda=1.474 #media de goles del R.Madrid
prob=(exp(-lambda)*lambda^x)/factorial(x)

y=0
lambda=1.632 #media de goles del Sevilla
prob=(exp(-lambda)*lambda^y)/factorial(y)
```

Se ha obtenido que $P(Real\ \ Madrid\ \ meta\ \ dos\ \ goles) = p(x = 2) = 0.2487797$. Mientras que $P(Sevilla\ \ no\ \ marque) = 0.1955381$. Como hemos asumido que son independientes $P(2-0) = P(x = 2) * P(y = 0) = 0.0486$

Este modelo introductorio presenta algunas carencias, por ejemplo no tiene en cuenta qué equipo juega en casa, esto podría subsanarse considerando dos distribuciones de Poisson para cada equipo, una para cuando sea local y otra para cuando sea visitante y en cada partido tomemos la adecuada para cada equipo.

Tampoco tiene en cuenta qué equipos se enfrentan así como que asume que las dos Poisson son independientes, es decir se asume que el número de goles de un equipo es independiente del del contrario. Por ello en la próxima sección

trataremos con un modelo más elaborado.

Vamos ahora a estudiar la evolución de los resultados jornada a jornada desde la 9 hasta la 19 y vamos a interesarnos principalmente en el número de aciertos en los goles de cada equipo individualmente. En esta ocasión sí usaremos la función "dpois" de R. Ahora sí distinguiremos local de visitante en nuestros datos.

Ejemplo 1.8. Para el partido de la jornada 9 entre Huesca y Espanyol se tiene que:
$\lambda_{\text{Huesca local}} = \frac{0+0+1+0}{4} = 0.25$

$\lambda_{\text{Espanyol visitante}} = \frac{1+1+0+2+2}{5} = \frac{6}{5}$

```
#La probabilidad del resultado
#que se dió en la realidad es
> dpois(0,0.25)
[1] 0.7788008
> dpois(2,1.2)
[1] 0.2168598
> dpois(0,0.25)*dpois(2,1.2)
[1] 0.1688906
#la probabilidad de goles de cada equipo se
#distribuye:
> dpois(0:10,0.25)
 [1] 7.788008e-01 1.947002e-01 2.433752e-02 2.028127e-03 ...
> dpois(0:10,1.2)
 [1] 3.011942e-01 3.614331e-01 2.168598e-01 8.674393e-02 ...
```

En esta ocasión lo más probable es que el Huesca marque 0 goles y el Espanyol marque 1. No es muy diferente del resultado final de 0-2 donde hemos acertado los goles del Huesca, y hemos fallado por la mínima los del Espanyol.

1.2.2 Modelo Poisson bivariante: Karlis y Ntzoufas (2003)

Propondremos en esta sección un nuevo modelo considerando dependencia entre los goles marcados por cada equipo en un partido, haremos un estudio análogo al anterior y analizaremos las diferencias.

Si GL y GV representan el marcador conseguido por cada uno de los equipos, una interpretación natural de los parámetros de un modelo bivariado de Poisson es que λ_1 y λ_2 reflejen la habilidad de marcar cada uno de los equipos y λ_3 refleje las condiciones del juego (por ejemplo, la velocidad del juego, el clima o

las condiciones del estadio).

El modelo en la observación i-ésima presenta la siguiente forma:

$$(GL_i, GV_i) \sim BP(\lambda_{1i}, \lambda_{2i}, \lambda_{3i})$$

$$\log(\lambda_{1i}) = \mu + hom + ath_i + deg_i$$

$$\log(\lambda_{2i}) = \mu + atg_i + deh_i$$

$$\log(\lambda_{3i}) = \beta_k + \sigma_1 \beta_1^{home} + \sigma_2 \beta_i^{away}$$

En las dos primeras ecuaciones μ es constante, hom es el parámetro que muestra el efecto de jugar en casa, at y de son el rendimiento ofensivo y defensivo respectivamente. Además g y h indican si el equipo i juega en casa en el primer caso, o fuera en el segundo.

En la última ecuación β_k es un parámetro común a todos los equipos, mientras que β_1^{home}, β_i^{away} son parámetros que indican si el equipo es local o visitante. Por último $\sigma_i \quad (i = 1, 2)$ es un indicador binario que depende del modelo que se considere:
Caso 1: $\sigma_1 = \sigma_2 = 0 \rightarrow$ Covarianza constante.
Caso 2: $\sigma_1 = 1, \sigma_2 = 0 \rightarrow$ La covarianza solo depende del equipo local.
Caso 3: $\sigma_1 = 0, \sigma_2 = 1 \rightarrow$ La covarianza solo depende del equipo visitante.
Caso 4: $\sigma_1 = \sigma_2 = 1 \rightarrow$ La covarianza depende de ambos equipos.

El efecto de λ_3 es la forma de actuar es aditiva sobre la media marginal y refleja las condiciones del juego.

Aplicación práctica del Algoritmo EM

En primer lugar hacemos una conversión para pasar de trabajar en tres variables a trabajar en dos. De este modo:

$$(X_{1i}, X_{2i}, X_{3i}) \overset{(}{\rightarrow} X_i = X_{1i} + X_{3i}, \quad Y_i = X_{2i} + X_{3i},$$

donde a la izquierda tenemos los datos observados y a la derecha los datos no

observados. Si estos últimos están disponibles la estimación es más sencilla. El procedimiento a seguir es ajustar los modelos de regresión de Poisson sobre X_1, X_2 y X_3. Debemos estimar las funciones de los datos no observados por esperanzas condicionadas y ajustar los modelos anteriores a los valores estimados en el paso E.

Definimos como $\phi = (\beta_1', \beta_2', \beta_3')$ el vector completo de parámetros estimados. Así el logaritmo de la ecuación de verosimilitud de los datos completos viene dada por:

$$\log(\phi) = -\sum_{i=1}^{n}\sum_{k=1}^{3}\lambda_{ki} + \sum_{i=1}^{n}\sum_{k=1}^{3}x_{ki}\log(\lambda_{ki}) - \sum_{i=1}^{n}\sum_{k=1}^{3}\log(x_{ki}!).$$

Véase que el último sumando no depende de ϕ, de modo que en términos de x_{ki} podemos decir que $\log(\phi)$ es una función lineal en términos de x_{ki}. Entonces el algoritmo EM procede de la siguiente manera:

Supongamos que estamos en la iteración k-ésima, entonces tendremos los valores $\delta^{(k)}$, $\lambda_{1i}^{(k)}$, $\lambda_{2i}^{(k)}$ y $\lambda_{3i}^{(k)}$. Calcularemos los valores esperados de X_{3i} aplicando el siguiente resultado:

Proposición 1.9. *Si* $\min(x_i, y_i) > 0$ *entonces*

$$s_i = E[X_{3i}|X_i, Y_i, \delta^{(k)}] = \lambda_{3i}^{(k)}\frac{f_{BP}(x_{i-1}, y_{i-1}|\lambda_{1i}^{(k)}, \lambda_{2i}^{(k)}, \lambda_{3i}^{(k)})}{f_{BP}(x_i, y_i|\lambda_{1i}^{(k)}, \lambda_{2i}^{(k)}, \lambda_{3i}^{(k)})}$$

Si $\min(x_i, y_i) = 0$ *entonces*

$$s_i = E[X_{3i}|X_i, Y_i, \delta^{(k)}] = 0.$$

Implementación en R

Vamos a utilizar como ejemplo la primera vuelta de LaLiga 2018-2019. El código en R es el siguiente:

```
#Primera vuelta temporada 2018/2019
datos1 = read.table("res_1_vuelta.txt",header=TRUE)
names(datos1)
attach(datos1)
datos1
levels(datos1[,3])
options(contrasts = c("contr.sum", "contr.poly"))
# Modelización de $\lambda_{1}$ y $\lambda_{2}
form1 <- ~c(team1,team2)+c(team2,team1)
# Modelo de la doble Poisson independiente
ex4.m1<-lm.bp( g1~1, g2~1, l1l2=form1, zeroL3=TRUE, data=datos1)
# Modelos de la Poisson bivariante para los diferentes casos
ex4.m2<-lm.bp(g1~1,g2~1, l1l2=form1, data=datos1)
ex4.m3<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1, data=datos1)
ex4.m4<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team2, data=datos1)
ex4.m5<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1+team2, data=datos1)

# Llamamos a los parámetros monitorizados por el primer modelo:
#Dbl Poisson ex4.m1$coef
# Todos los parámetros ex4.m1$beta1
# Parámetros del modelo para $\lambda1$ ex4.m1$beta2
# análogo para $\lambda_{2}
# Todos son iguales que en $\beta1$ excepto el intercepto
ex4.m1$beta2[1] # Intercepto para $\lambda_{2}$
ex4.m1$beta1[1]-ex4.m1$beta2[1] # efecto local estimado
#Coeficiente del vigésimo equipo en ataque  alfabéticamente
-sum(ex4.m1$coef[2:20])
# Análogamente para el nivel defensivo(team2..team1)
-sum(ex4.m1$coef[21:39])
# mostrar los parámetros para el modelo 2 BivPoisson(lamdba1,lambda2,
constant lamdba3)
ex4.m2$beta1 # Parámetros del modelo para $\lambda1$
ex4.m2$beta2 # Análogo para $\lambda_{2}
# Todos son iguales en $\beta_{1}$ excepto el intercepto
ex4.m2$beta3 # Parámetros $\lambda_{3}
exp(ex4.m2$beta3)
ex4.m2$beta2[1] # Intercepto para $\lambda2$.
ex4.m2$beta1[1]-ex4.m2$beta2[1] # efecto jugar en casa
# Potencial atacante del vigésimo equipo (team1..team2)

-sum(ex4.m2$coef[ 2:20])
# Análogamente para el potencial defensivo (team2..team1)
-sum(ex4.m2$coef[21:39])
```

```
names(ex4.m1)
ex4.m1$coefficients
ex4.m1$fitted.values
ex4.m1$residuals
ex4.m1$beta1
ex4.m1$beta2
ex4.m1$lambda1
ex4.m1$lambda2
ex4.m1$lambda3
ex4.m1$loglikelihood
ex4.m1$iterations
ex4.m1$parameters
ex4.m1$AIC
ex4.m1$BIC
names(ex4.m2) #Idem para ex4.m3;ex4.m4;ex4.m5;ex4.m6;ex4.m7;
ex4.m8;ex4.m9;ex4.m10;ex4.m11;ex4.m12
```

Con este código estimamos los parámetros del modelo de Karlis y Ntzoufras, en la próxima sección procederemos al análisis de datos.

Análisis de datos

Vamos a proceder con el análisis de nuestros datos, esto es, la primera vuelta de la liga 2018-19. Vamos a utilizar el código visto al final del apartado anterior e iremos explicando el proceso computacional con más detenimiento.

Lo primero que hacemos es utilizar un fichero que contenga los resultados de los 190 partidos de la primera vuelta de LaLiga del presente año y así posibilitar la lectura en R. Ponemos las primeras líneas de un fichero para hacerlo más visual:

```
jornada g1 g2 team1 team2
1 0 0 GIR VLL
1 0 3 BET LEV
1 1 1 CEL ESP
1 1 2 VIL RSO
```

```
> datos1 <- read.table("LaLiga18-19.txt",header=TRUE)
```

Utilizaremos cinco variables que serán *gl*=Goles local, *gv*=goles visitante *team1* y *team2*, la unión de estas constituye nuestro conjunto de datos. Además,

las mismas serán clave en la definición de nuestras variables explicativas, que son *att* y *def* (potencial en ataque y potencial en defensa respectivamente).

```
> names(datos1)
#ejecuto
[1] "g1" "g2" "team1" "team2"
> datos1
```

Los niveles listan los 20 equipos de la primera división durante esta temporada:

```
> levels(datos1[,3])
[1]"Alavés"  "Athletic" "AtMadrid" "Barcelona" "Eibar"

 [6]"Espanyol" "Getafe"  "Girona"  "Leganés"  "Levante"

[11]"Rayo Vallecano" "R.Betis"  "R.Celta"  "RMadrid"

   [16]"RSociedad"  "RValladolid"  "S.D.Huesca"  "Sevilla"

"Valencia" "Villarreal"
```

En este modelo se supone que *att* y *deff* para cada equipo es independiente del adversario. A continuación vamos a modelar los efectos comunes de ataque y defensa. Lo haremos con un vector $c(team1, team2)$ para el ataque, y $c(team2, team1)$ para la defensa. De este modo ya es posible modelizar λ_i $i = 1, 2$.

```
 form1<-c(team1,team2)+c(team2,team1)
 #codif. variables explicativas
options(contrast=c("contr.sum","contr.poly"))
```

Así se fuerza que la suma de coeficientes sea 0. Restricción de la que hablamos en el marco teórico. La interpretación de cada coeficiente es la desviación de la capacidad ofensiva (respectivamente defensiva) de cada equipo respecto de las de un equipo con un potencial ofensivo (respectivamente defensivo) medio.

Procede ahora ir al ajuste mediante la función lm.bp. Lo haremos para cinco modelos. En primer lugar, el modelo de la Poisson bivariada considerando $\lambda_3 = 0$, y luego los otros cuatro modelos, cada uno de los cuales se diferencia en la forma de tomar de λ_3, estos son: constante, dependiente del local (r. visitante) o de ambos.

Podemos llamar a las funciones de la siguiente forma:

```
ex4.m1<-lm.bp( g1~1, g2~1, l1l2=form1, zeroL3=TRUE, data=datos1)
ex4.m2<-lm.bp(g1~1,g2~1, l1l2=form1, data=datos1)
ex4.m3<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1, data=datos1)
ex4.m4<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team2, data=datos1)
ex4.m5<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1+team2, data=datos1)
```

Estas funciones nos dan una lista de valores que definimos en la sección teórica *(coefficients, fitted.values, residuals, loglikelihood, parameters, AIC BIC ...)* Destaquemos ahora los dos últimos objetos nombrados de esta lista, el criterio de información de Akaike AIC y el criterio de información de Bayes. La interpretación es sencilla, simplemente consideraremos mejores modelos los que tengan valores de AIC y BIC más bajos.

Tomaremos el mejor modelo en términos de AIC y BIC. Para ver la aplicación es conveniente verlo en dos ejemplos reales, estos serán la primera vuelta de La Liga Española de primera división, tanto en la sección masculina como femenina.

En el apartado anterior hemos visto la mayor parte del código utilizado. Vamos a verlo aplicado a este ejemplo concreto:

```
#análisis de datos para la primera vuelta de LaLiga 18-19
library(bivpois)
muestra = read.table("LaLiga18-19.txt",header=TRUE)
names(muestra)
muestra
levels(muestra[,3])
options(contrasts = c("contr.sum", "contr.poly"))
# Modelización de $\lambda_{1}$ y $\lambda_{2}
form1 <- ~c(team1,team2)+c(team2,team1)
```

En esta primera parte del código, llamamos a la librería *bivpois* para poder utilizar sus funciones. A continuación definimos como *muestra* al fichero de datos que ha sido cargado y mostramos los niveles, esto es, la nomenclatura de los equipos en orden alfabético. Finalmente ordenamos la codificación de las variables propuesta por los autores.

```
# Modelo de la doble Poisson independiente
ex4.m1<-lm.bp( g1~1, g2~1, l1l2=form1, data=muestra)
# Modelos de la Poisson bivariante para los diferentes casos
ex4.m2<-lm.bp(g1~1,g2~1, l1l2=form1, data=muestra)
ex4.m3<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1, data=muestra)
ex4.m4<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team2, data=muestra)
ex4.m5<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1+team2, data=muestra)
```

```
# Elección del mejor modelo
ex4.m1$AIC
ex4.m1$BIC
ex4.m1$parameters
ex4.m2$AIC
ex4.m2$BIC
ex4.m2$parameters
ex4.m3$AIC
ex4.m3$BIC
ex4.m3$parameters
ex4.m4$AIC
ex4.m4$BIC
ex4.m4$parameters
ex4.m5$AIC
ex4.m5$BIC
ex4.m5$parameters
```

Utilizamos ahora los modelos correspondiendes a la función lm.bp perteneciente a la librería bivpois. Cada uno de los 5 que hemos cargado nos devolverá una lista de objetos, los cuales hemos enunciado en la sección bivpois y la función lm.bp. En particular, en este momento estamos interesados en los coeficientes AIC y BIC y así determinar el mejor modelo para la presente muestra. Este ha sido el resultado:

Modelo	AIC	BIC
Doble Poisson	$1092,86$	$1254,41$
Poisson Bivariada	$1092,86$	$1254,41$
Poisson Bivariada Local	$1112,58$	$1348,99$
Poisson Bivariada Visitante	$1113,03$	$1349,41$
Poisson Bivariada Ambos	$1121,48$	$1432,76$

Buscamos los valores más bajos de AIC y BIC, luego el mejor modelo es el de la Poisson Bivariada con $\lambda_3 = 0$. Así, proseguiremos usando dicho modelo.

Una vez ejecutado el modelo tomamos el modelo de la poisson bivariante (modelo 2) con las siguientes secuencias podemos ver los objetos que nos da la funcion *lm.bp*.

```
names(ex4.m2)
ex4.m2$coefficients
ex4.m2$fitted.values
ex4.m2$residuals
ex4.m2$beta1
```

```
ex4.m2$beta2
ex4.m2$beta3
ex4.m2$lambda1
ex4.m2$lambda2
ex4.m2$lambda3
ex4.m2$loglikelihood
ex4.m2$iterations
```

Si ejecutamos las ordenes anteriores podremos ver los valores para cada equipo de cada objeto de la lista:

Usaremos como ejemplo *coefficients* para hacer una observación acerca de la codificación.

```
> ex4.m2$coefficients
   (l1):(Intercept)  (l1):team1..team21 (l1):team1..team210
       -0.055030568         0.012202695       -0.115338652
```

Tenemos el intercepto y el coeficiente en ataque de tres equipos que podemos reconocer gracias al dígito que sigue de *team1..team2*, en este caso son 1 y 11, estos números se corresponden con el orden de nivel (alfabético). Es decir son el Alavés y Girona respectivamente. También nos dan los parámetros de defensa, veremos ahora otro fragmento para verlo en *R*:

```
(l2):team2..team12  (l2):team2..team13  (l2):team2..team14
       0.021168562        -1.008673946        -0.103949969
```

De manera análoga a lo anterior, se tiene que este es el parámetro defensivo de Athletic, Atlético y Barcelona. Sin embargo falta un parámetro defensivo y ofensivo, este es el del último nivel, es decir el último equipo alfabéticamente hablando, que es el Valladolid (codificado bajo las siglas "VLL"). Para calcular estos coeficientes podemos utilizar que los definimos tal que la suma de todos sea nula, tanto en ataque como en defensa. Así:

```
> -sum(ex4.m2$coef[ 2:20])
[1] -0.6356465
> # Análogamente para el potencial defensivo de dicho equipo
> -sum(ex4.m2$coef[21:39])
[1] -0.2993231
```

Luego ya tenemos una tabla con los potenciales ofensivos y defensivos de los 20 equipos de primera división basándonos en los datos de la primera vuelta:

Equipo	Ataque	Defensa
Alavés	0,01	-0,2
Athletic	-0,25	0,02
Atlético	0,1	0,3
Barcelona	0,99	-0,1
Betis	-0,02	0,05
Celta	0,37	0,39
Eibar	-0,09	0,25
Espanyol	-0,01	0,26
Getafe	0,01	-0,34
Girona	-0,12	-0,01
Huesca	-0,42	0,51
Leganés	-0,53	-0,24
Levante	0,41	0,56
Rayo V.	0,02	0,54
Real Madrid	0,37	0,16
Real Sociedad	0,00	0,07
Sevilla	0,42	-0,11
Valencia	-0,43	-0,44
Villareal	-0,19	0,08
Valladolid	-0.64	-0.30

Los $\lambda_i \quad i = 1, 2$ han sido calculados de la forma:

$$\log(\lambda_{1i}) = \mu + hom + at_{hi} + de_{gi}$$

$$\log(\lambda_{2i}) = \mu + at_{gi} + de_{hi}$$

a partir de los coeficientes anteriores:

Para La Liga Iberdrola, de manera análoga elegimos el mejor modelo.

```
library(bivpois)
datos1 = read.table("iberdrola18-19.txt",header=TRUE)
names(datos1)
datos1
levels(datos1[,3])
options(contrasts = c("contr.sum", "contr.poly"))
# Modelización de $\lambda_{1}$ y $\lambda_{2}
form1 <- ~c(team1,team2)+c(team2,team1)
# Modelo de la doble Poisson independiente
```

```
ex4.m1<-lm.bp( g1~1, g2~1, l1l2=form1, data=datos1)
# Modelos de la Poisson bivariante para los diferentes casos vistos
ex4.m2<-lm.bp(g1~1,g2~1, l1l2=form1, data=datos1)
ex4.m3<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1, data=datos1)
ex4.m4<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team2, data=datos1)
ex4.m5<-lm.bp(g1~1,g2~1, l1l2=form1, l3=~team1+team2, data=datos1)
ex4.m1$AIC
ex4.m1$BIC
ex4.m1$parameters
ex4.m2$AIC
ex4.m2$BIC
ex4.m2$parameters
ex4.m3$AIC
ex4.m3$BIC
ex4.m3$parameters
ex4.m4$AIC
ex4.m4$BIC
ex4.m4$parameters
ex4.m5$AIC
ex4.m5$BIC
ex4.m5$parameters
```

Podemos plasmar el resultado en la siguiente tabla:

Modelo	AIC	BIC
Doble Poisson	$732,8$	$847,67$
Poisson Bivariada	$732,8$	$847,7$
Poisson Bivariada Local	$749,41$	$916,48$
Poisson Bivariada Visitante	$748,68$	$915,74$
Poisson Bivariada Ambos	$760,65$	$979,63$

De donde concluimos que los mejores modelos son el *m1* y el *m2*. Procederemos con el último. Con el siguiente código obtenemos todos los objetos de la función *lm.bp* para *m2*.

```
names(ex4.m2)
ex4.m2$coefficients
ex4.m2$fitted.values
ex4.m2$residuals
ex4.m2$beta1
ex4.m2$beta2
ex4.m2$beta3
ex4.m2$lambda1
ex4.m2$lambda2
```

```
ex4.m2$lambda3
ex4.m2$loglikelihood
ex4.m2$iterations
# monitorización de parámetros para el modelo 2
ex4.m2$beta1 # Parámetros del modelo para $\lambda1$
ex4.m2$beta2 # Análogo para $\lambda_{2}
# Todos son iguales en $\beta_{1}$ excepto el intercepto
ex4.m2$beta3 # Parámetros $\lambda_{3} (En este caso solo el intercepto)
ex4.m2$lambda3
exp(ex4.m2$beta3)
ex4.m2$beta1[1]#Intercepto para $\lambda1$.
ex4.m2$beta2[1] # Intercepto para $\lambda2$.
ex4.m2$beta1[1]-ex4.m2$beta2[1] # efecto jugar en casa
# Potencial atacante del vigésimo equipo alfabéticamente
#(team1..team2)

-sum(ex4.m2$coef[ 2:16])
# Análogamente para el potencial defensivo de dicho equipo
-sum(ex4.m2$coef[17:33])
```

Equipo	Ataque	Defensa
Albacete	0,24	0,6
Athletic	0,22	-0,46
Atlético	0,9	0,64
Barcelona	0,86	-1,48
Betis	0,18	-0,12
Espanyol	0,7	0,14
Granadilla	0,07	0,3
Spo. Huelva	-0,6	0,02
Levante	0,52	-0,78
Logroño	-0,26	0,38
Madrid CFF	-0,16	0,63
Málaga	-0,62	0,54
Rayo V.	-0,09	0,12
R. Sociedad	0,1	0,14
Sevilla	-0,33	-0,55
Valencia	-0,33	-0,17

Ejemplos

Con las tablas de Items obtenidas podemos estimar la probabilidad para un partido de la jornada 20 de LaLiga o de la jornada 16 de La Liga Iberdrola, en ambos casos la primera de la segunda vuelta. Para el resto del campeonato

en ambos ámbitos habría que añadir las jornadas disputadas al fichero.txt. Por ejemplo, para un partido de la jornada 27, tendríamos que tener un fichero con las 26 primeras jornadas. Los datos expuestos en este trabajo abarcan la primera vuelta de las dos ligas mencionadas, de modo que haremos ejemplos para la jornada 20 de LaLiga, y de la jornada 16 para La Liga Iberdrola.

Como tenemos un valor de λ para cada observación (partido) usaremos el más reciente teniendo en cuenta si se juega en casa o fuera.

Implementaremos en R la función de probabilidad de la Poisson bidimensional para hacer los cálculos, pondremos los valores numéricos de uno de los ejemplos:

```
lambda1=3.266
lambda2=0.738
lambda3=0.282
x=2
y=1 #estos cinco valores dependen del ejemplo dado
datos=c(lambda1,lambda2)
a=exp(-lambda1-lambda2-lambda3)*lambda1^x*lambda2^y/

factorial(x)*factorial(y)
kmin=min(x,y)
suma=0
for (k in 0:kmin) {
  suma=suma+(choose(x,k)*choose(y,k)*factorial(k)*(lambda3/

(lambda1*lambda2))^k)
}
b=suma
probabilidad=a*b
```

Ejemplo 1.10. Para el partido de la jornada 20 de LaLiga entre el Barcelona y el Leganés tenemos los siguiente valores: $\lambda_1 = 3.266$ $\lambda_2 = 0.738$ $\lambda_3 = 0.282$. Con esto podemos calcular la probabilidad de los diferentes resultados. Ponemos 3 como ejemplo:

Para el resultado 2-1 se toma (x,y)=(2,1)

$$f_{BP}(2,1) = P(X = x, Y = y) = e^{-(\lambda_1+\lambda_2+\lambda_3)} \frac{\lambda_1^2}{2!} \frac{\lambda_2^1}{1!} \sum_{k=0}^{\min(1,2)} \binom{2}{k} \binom{1}{k} k! (\frac{\lambda_3}{\lambda_1 \lambda_2})^k = 0.07$$

Para el resultado 0-0 se toma (x,y)=(0,0)

$$f_{BP}(0,0) = P(X = 0, Y = 0) = e^{-(\lambda_1+\lambda_2+\lambda_3)} \frac{\lambda_1^0}{0!} \frac{\lambda_2^0}{0!} \sum_{k=0}^{\min(0,0)} \binom{0}{k} \binom{0}{k} k! (\frac{\lambda_3}{\lambda_1 \lambda_2})^k = 0.01$$

Para el resultado 0-1 se toma (x,y)=(0,1)

$$f_{BP}(0,1) = P(X=0,Y=1) = e^{-(\lambda_1+\lambda_2+\lambda_3)}\frac{\lambda_1^0}{0!}\frac{\lambda_2^1}{1!}\sum_{k=0}^{\min(0,1)}\binom{0}{k}\binom{1}{k}k!(\frac{\lambda_3}{\lambda_1\lambda_2})^k = 0.01$$

Para el partido de la jornada 26 de La Liga Iberdrola entre el Atlético y el Athletic tenemos los siguiente valores: $\lambda_1 = 3.23$ $\lambda_2 = 1.35$ $\lambda_3 = 0.09$. Vamos a calcular la probabilidad de algunos resultados arbitrarios mediante la función de probabilidad:

$$P(2-2) = P(X=2,Y=2) = 0.1930682$$
$$P(3-1) = P(X=3,Y=1) = 0.07546168$$
$$P(0-2) = P(X=0,Y=2) = 0.03416192$$

Vamos a proceder a hacer un estudio análogo al de la sección anterior, tomando ahora este último modelo. Proseguimos para todos los partidos de nuestra muestra tal como en el ejemplo que mostramos a continuación. Usaremos la fórmula:

$$dpois(0:10,\lambda_1+\lambda 3)\cdot dpois(y,\lambda_2+\lambda 3)$$

Con esta calculamos la probabilidad de cualquier resultado dentro del estilo del quinigol. siendo M la suma de las componentes del vector que devuelve $dpois(3:10,\lambda_1+\lambda 3)\cdot dpois(y,\lambda_2+\lambda 3)$

Ejemplo 1.11. Para el partido de la jornada 9 entre el Eibar y el Athletic se tiene que:

```
ex4.m2<-lm.bp(g1~1,g2~1, l1l2=form1, data=muestra[1:80,])
#leo los primeros 80 partidos es decir, las 8 primeras jornadas
> ex4.m2$lambda1[65]
       65
0.7207508
> ex4.m2$lambda2[64]
     144
0.9623561
```

buscando el número del partido en el fichero obtenemos el λ_1 más reciente para el Eibar y el λ_2 más reciente para el Athletic.

```
> ex4.m2$lambda3[1]
        1
0.0574329
#Recordemos que lambda3 es constante
```

Podemos proceder ahora a calcular la Poisson bidimensional correspondiente a estos datos para ver la distribución del marcador.

```
> c(dpois(0:2, 0.72+0.06),1-ppois(2, 0.72+0.06))
[1] 0.45840601 0.35755669 0.13944711 0.04459019
> c(dpois(0:2, 0.96+0.06),1-ppois(2, 0.96+0.06))
[1] 0.36059494 0.36780684 0.18758149 0.08401673
```

Generamos un vector con las probabilidades de 0,1,2 y M respectivamente para el equipo local y para el visitante.

```
> outer(c(dpois(0:2, 0.72+0.06),1-ppois(2, 0.72+0.06)),

 c(dpois(0:2, 0.96+0.06),1-ppois(2, 0.96+0.06)),"*")
           [,1]       [,2]        [,3]        [,4]
[1,] 0.16529889 0.16860487 0.085988482 0.038513775
[2,] 0.12893313 0.13151180 0.067071016 0.030040745
[3,] 0.05028392 0.05128960 0.026157696 0.011715890
[4,] 0.01607900 0.01640058 0.008364294 0.003746322
```

Obtenemos la matriz de probabilidades para cada resultado. Concretamente esta es:

Goles	0	1	2	M
0	0.165	0.169	0.086	0.039
1	0.129	0.132	0.067	0.030
2	0.050	0.051	0.026	0.011
M	0.016	0.016	0.08	0.003

Las columnas representan al que juega de local y las filas al visitante.

El resultado más probable es el 1-1 con una probabilidad de 0.169, seguido muy de cerca por el 0-0, con una probabilidad del 0.165. precisamente 1-1 fue el resultado real del partido.

1.3 Análisis de la calidad de los modelos

En esta sección veremos la cantidad media de aciertos en las muestras que estamos estudiando para los dos modelos desarrollados en esta memoria.

1.3.1 Calidad del modelo de Poisson en una variable

```
marcadores <- read.table("LaLiga18-19.txt",header=TRUE)
names(marcadores)
jornadasestudio <- 9:19
aciertospartido <- numeric(length(jornadasestudio))
names(aciertospartido) <- jornadasestudio
aciertosequipo  <- numeric(length(jornadasestudio))
names(aciertosequipo) <- jornadasestudio
```

Comenzamos leyendo el fichero de datos y definimos las variables a utilizar:

jornadasestudio: jornadas que hemos analizado anteriormente para asignar una distribución de probabilidad a sus partidos.

aciertospartido: número medio de resultados acertados

aciertosequipo: número medio de goles de cada equipo (individualmente) acertados.

```
for (jor in jornadasestudio){
  rj = marcadores[marcadores$jornada == jor,] #jornada jor
  ra = marcadores[marcadores$jornada < jor,]  #jornadas anteriores
  totalprbmarcador <- 0
  totalprbgoles <- 0
  numpartidos <- dim(rj)[1]
```

Generamos un bucle para recorrer *jornadasestudio*, en este usaremos *rj*, donde almacenaremos los partidos de la jornada correspondiente a la iteración en curso, así como con *ra* guardaremos el registro de las jornadas anteriores.

```
  for (i in 1:numpartidos){
    eqlocal <- rj$team1[i]
    eqvisit <- rj$team2[i]
    lambda1 <- mean(ra$g1[ra$team1 == eqlocal])
    lambda2 <- mean(ra$g2[ra$team2 == eqvisit])
    probg1 <- dpois(rj$g1[i], lambda = lambda1)
    probg2 <- dpois(rj$g2[i], lambda = lambda2)

    totalprbmarcador <- totalprbmarcador + probg1 * probg2
```

Probabilidad acertar el marcador:

```
    totalprbgoles <- totalprbgoles + probg1 + probg2
```

Suma de probabilidad de acertar los goles de los equipos:

```
    }
    aciertospartido[as.character(jor)] <- totalprbmarcador/numpartidos
    aciertosequipo[as.character(jor)] <- totalprbgoles/numpartidos/2
  }
```

Dentro del bucle anterior inicializamos otro bucle que recorre todos los partidos de nuestros datos, una vez dentro debemos calcular el λ_1 y λ_2 para todos los partidos, donde para el primero es la media de goles en casa hasta el momento del equipo local y el segundo es la media de goles fuera de casa para el visitante. Utilizamos esos valores para para calcular las Poisson que da la distribución de probabilidad para los goles del equipo local y visitante. Les hemos asignado los nombres de *probg1* y *probg2* respectivamente.

Utilizaremos *aciertospartido* para calcular la esperanza de los aciertos, esto lo hacemos acumulando la probabilidad de acertar cada partido. Análogamente lo hacemos para los equipos por separado con *aciertosequipo*.

```
maxy <- max(aciertosequipo)*1.1
plot(names(aciertosequipo), aciertosequipo, type='l',

 ylim=c(0,maxy), xlab='jornada', ylab='prob. aciertar',
     main='Probabilidades de acertar goles o resultado')
lines(names(aciertospartido), aciertospartido, col='blue')
text(18,0.25,labels='goles del equipo')
text(18,0.1,labels='resultado partido', col='blue')
```

Finalmente generamos una gráfica con la evolución del número medio de aciertos tanto de goles de equipos como de resultados acertados.

Se observa que con este modelo para la muestra de LaLiga 18-19 tiene un número de aciertos medio que varía entre el 5% y el 10%. Mientras para los goles de cada equipo, tenemos una precisión que se mueve entre el 25% y 30% de éxitos.

Si repetimos este mismo proceso con La Liga Iberdrola entre la jornada 9 y la 16 tenemos el siguiente resultado:

Se aprecia una precisión media que varía entre un 3% y un 8% en cuanto al marcador, mientras que para los goles medios de cada equipo en un partido individualmente este rango se mueve entre el 21% y el 28%.

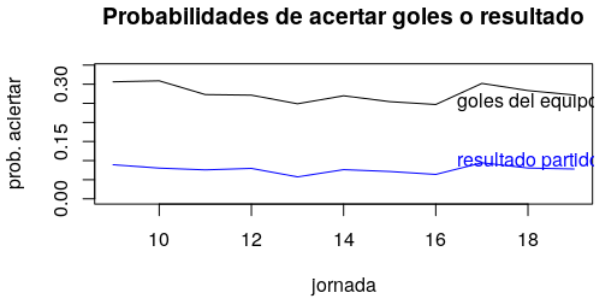

Fig. 1.1. Calidad del modelo independiente para LaLiga

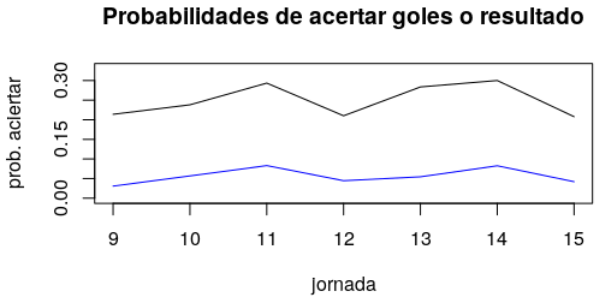

Fig. 1.2. Calidad del modelo independiente para La Liga Iberdrola

1.3.2 Calidad del modelo de Karlis y Ntzoufras (2003)

Nos cuestionamos ahora si considerando que los goles locales y visitantes tienen una correlación podemos mejorar estas predicciones. Para ello vamos a realizar el mismo análisis en R, donde la única diferencia que tiene el algoritmo que usaremos ahora con el caso independiente es la forma de calcular las $\lambda_i \quad i = 1, 2, 3$, lo haremos mediante el algoritmo propuesto por Karlis y Ntzoufras.

```
#análisis de datos para la primera vuelta de LaLiga 18-19
library(bivpois)
marcadores = read.table("LaLiga18-19.txt",header=TRUE)
names(marcadores)
marcadores
levels(marcadores[,4])
```

```
#rj serán todos los partidos de la jornada jor
options(contrasts = c("contr.sum", "contr.poly"))
# Modelización de $\lambda_{1}$ y $\lambda_{2}
form1 <- ~c(team1,team2)+c(team2,team1)
```

Observamos que se programa la modelización de las $\lambda_i, (i = 1, 2, 3)$ tal como proponen los autores.

```
jornadasestudio <- 9:19
aciertospartido <- numeric(length(jornadasestudio))
names(aciertospartido) <- jornadasestudio
aciertosequipo  <- numeric(length(jornadasestudio))
names(aciertosequipo) <- jornadasestudio

for (jor in jornadasestudio){
  rj = marcadores[marcadores$jornada == jor,]
  ra = marcadores[marcadores$jornada < jor,]
  totalprbmarcador <- 0
  totalprbgoles <- 0
  numpartidos <- dim(rj)[1]
  ex4.m2<-lm.bp(g1~1,g2~1, l1l2=form1, data=ra)
  ra[,'lambdal'] <- ex4.m2$lambda1 + ex4.m2$lambda3
  ra[,'lambdav'] <-ex4.m2$lambda2 + ex4.m2$lambda3
```

Implementamos el modelo correspondiente y hacemos una conversión para calcular λ_l y λ_v de tal manera que tenemos dos Poissones independientes para cada equipo en cada partido, cuya distribución de probabilidad es la misma que la de la Poisson Bivariante. El bucle que sigue está dentro del anterior:

```
  for (i in 1:numpartidos){
    eqlocal <- rj$team1[i]
    eqvisit <- rj$team2[i]
    lambda1 <- tail(ra$lambdal[ra$team1 == eqlocal], n=1)
    lambda2 <- tail(ra$lambdav[ra$team2 == eqvisit], n=1)
    probg1 <- dpois(rj$g1[i], lambda = lambda1)
    probg2 <- dpois(rj$g2[i], lambda = lambda2)
    totalprbmarcador <- totalprbmarcador + probg1 * probg2
    totalprbgoles <- totalprbgoles + probg1 + probg2
  }
  aciertospartido[as.character(jor)] <- totalprbmarcador/numpartidos
  aciertosequipo[as.character(jor)] <- totalprbgoles/numpartidos/2
```

```
}
maxy <- max(aciertosequipo)*1.1
plot(names(aciertosequipo), aciertosequipo, type='l',

ylim=c(0,maxy), xlab='jornada', ylab='prob. aciertar',
     main='Probabilidades de acertar goles o resultado')
lines(names(aciertospartido), aciertospartido, col='blue')
text(18,0.25,labels='goles del equipo')
text(18,0.1,labels='resultado partido', col='blue')
```

Como observamos el resto del código es análogo al problema independiente. Este es el resultado obtenido:

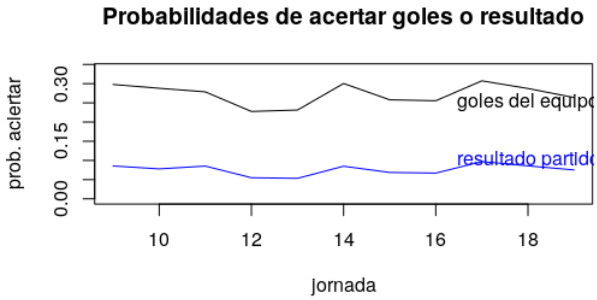

Fig. 1.3. Calidad del modelo bidimensional para LaLiga

Adjuntamos también una gráfica conjunta de LaLiga con ambos modelos, hacemos lo mismo con La Liga Iberdrola.

Nota: las representaciones gruesas identifican los modelos bivariantes, mientras que las más finas representan los univariantes.

Los aciertos a nivel de resultado varían cada jornada en el intervalo 5% y 10%, mientras que para los equipos tenemos un mínimo del 22% y un máximo del 30%, haciendo para esta muestra de LaLiga, que el modelo independiente sea escasamente mejor en general.

Cambiando el fichero de datos y usando el de La Liga Iberdrola el resultado es:

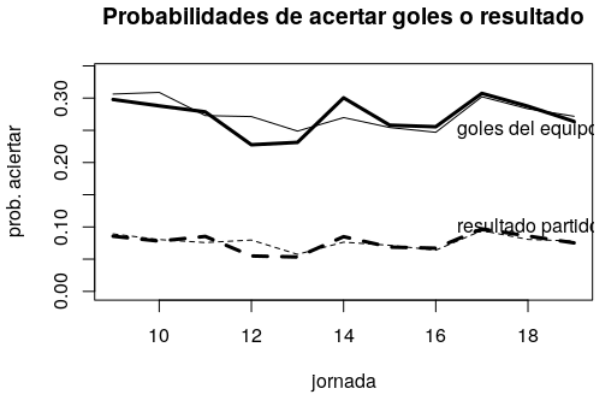

Fig. 1.4. Comparación de modelos para LaLiga 18-19

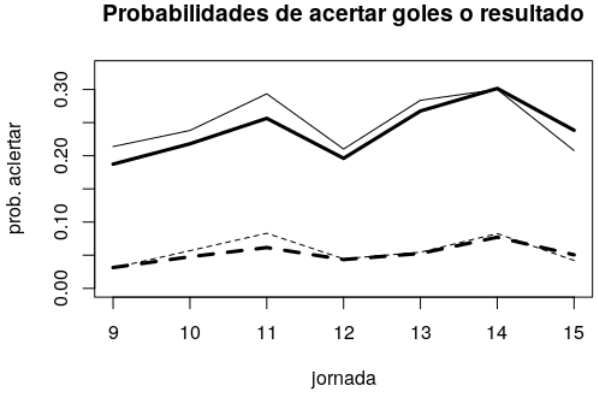

Fig. 1.5. Comparación de modelos para La Liga Iberdrola 18-19

Hay una precisión media de éxito en los resultados que varía entre un 3% y un 6%. Para cada equipo está entre un 17% y un 26%.

Existen casos (jornadas) en ambas muestras en las que el modelo bivariante mejora al independiente, sin embargo, la media general dictamina que en ambos

casos este último modelo es más preciso en los datos que hemos estudiado, que es medir el éxito medio de resultados y goles de un equipo en cada partido.

2

Cadenas de Markov

2.1 Marco teórico

Las cadenas de Markov son un tipo de modelo probabilístico en el que un sistema va cambiando su estado en el tiempo. Estos cambios de estado se asume que se realizarán de forma aleatoria e independiente de la historia completa: lo único importante es el estado actual.

Hemos desarrollado el marco teórico con la ayuda del post [4]

2.1.1 Nociones básicas

Sistema: Es el modelo del experimento que vamos a llevar a cabo, donde consideramos m estados denotados como E_i con $i = 1, 2, \dots m$. Donde cada uno lleva asociada una probabilidad.

Cadena de Markov: Es un proceso discreto, consiste una secuencia de estados en la que la variable aleatoria X_n va cambiando de estado en cada instante de tiempo.

Las cadenas de Markov cumplen la siguiente propiedad:

Proposición 2.1. *(Propiedad de Markov): $P(X_n = j)$ únicamente depende del estado anterior del sistema, es decir X_{n-1}.*

2.1.2 Probabilidades de transición

Definimos p_{ij} como la probabilidad de pasar del estado E_i al estado E_j con $i, j = 1, 2, 3, \dots, m$. Es decir:

$$p_{ij} = P(X_n = j | X_{n-1} = i)$$

Decimos que si el estado $p_{ij} > 0$ se tiene que E_i se puede comunicar con E_j. Sin embargo, esto no implica necesariamente que $p_{ji} = 0$. En el caso de que ambos se cumplen decimos que hay comunicación mutua.

Sea i fijo, se dice que los $p_{ij}, (j = 1, 2, \ldots, m)$ es una distribución de probabilidad ya que $\sum_{j=1}^{m} p_{ij} = 1$.

Vamos a definir una matriz que contenga las probabilidades de todos los casos posibles de transición p_{ij} para todo $i, j = 1, 2, \ldots, m$. Se la denominará matriz estocástica y la denotaremos como T.

$T \in \mathrm{M}_{mxm}$ tal que $T = [p_{ij}]$ tal que

$$T^{m \times m} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{pmatrix}$$

Se tiene que si las matrices $A = [a_{ij}]$ y $B = [b_{ij}]$ son matrices estocásticas, entonces Si $C = [c_{ij}]$ de modo que $C = A * B$, también será matriz estocástica.

Una consecuencia importante de la propiedad anterior es que si T es estocástica, entonces T^n será estocástica para todo n $\in \mathbb{N}$.

2.1.3 Tipos de probabilidades en una cadena de Markov

Definimos $p_j^{(n)}$ como la probabilidad de llegar a j en n pasos. En consecuencia, $p_j^{(0)}$ es la probabilidad de que el estado inicial sea p_j.

Denominamos $p^{(0)} = (p_1^{(0)}, \ldots, p_m^{(0)})$. En general $P^{(i)} = (p_1^{(i)}, \cdots, p_m^{(i)})$ es la distribución de probabilidad de llegar al estado i en j pasos con $j = 1, 2, \ldots, m$.

La utilidad de esta notación es que se tiene que:

$$P^{(1)} = (p_j^{(1)}) = \sum_{i=1}^{m} P^{(0)} p_{ij} = P^{(0)} T$$

En general:

$$P^{(n)} = (p_j^{(n)}) = P^{(n)}T$$

A la notación anterior podemos añadirle la condición de que empezamos en el estado i esto se denota $p_{ij}^{(n)}$.

Proposición 2.2. *(Ecuación de Kolmogorov):*

$$p_{ij}^{(n)} = \sum_{k=1}^{m} p_{ik}^{(n-1)} p_{kj}^{(1)} = \sum_{k=1}^{m} p_{ik}^{(n-1)} p_{kj}^{(1)}.$$

Observación:

$$[p_{ij}^{(n)}] = [p_{ik}^{(n)} p_{kj}^{(n)}] = T^n.$$

2.1.4 Tipos de estado en una cadena de Markov

A continuación hablaremos de los posibles estados en una cadena de Markov:

Estado absorbente: es aquel en el que si se entra, entonces no se puede salir. Esto es $p_{ii} = 1$ y $p_{ij} = 0$ para todo $i, j = 1, 2, \ldots, m \quad \text{con} \quad i \neq j$.

Estado periódico: Sea $t \in \mathbb{N}$, decimos que el estado E_i es periódico si fijado t, se tiene que $p_{ii}^{(n)} \neq 0$ para cada $n = mt$ con $m \in \mathbb{N}$. Es decir, múltiplos de t.

Estado aperiódico: E_i lo es si no es periódico. Es decir, para todo $t \in \mathbb{N}$ tal que si $n = mt$ con $m \in \mathbb{N}$, se tiene que $p_{ii}^{(n)}0$.

Estos dos últimos estados se pueden definir alternativamente. Si definimos $d(i) = mcd(n \quad | \quad p_{ii}^{(n)} > 0)$. Decimos que un estado es periódico si $d(i) > 1$, mientras que si $d(i) = 1$ diremos que es aperiódico.

Estado recurrente: Para este estado nos definimos f_j^n como la probabilidad de que la primera visita al estado E_j se produzca en el paso n. Nótese la distinción con p_{jj}^n ya que esta es independiente de que dicho estado haya sido visitado en alguna etapa anterior a n. De hecho la relación es la siguiente:

$$p_{jj}^{(n)} = f_j^n + \sum_{k=1}^{n-1} f_j^k p_{jj}^{(n-k)}$$

Como observación, la igualdad anterior se puede expresar en términos de f_j de la siguiente forma:

$$f_j^n = p_{jj}^{(n)} - \sum_{k=1}^{n-1} f_j^k p_{jj}^{(n-k)}$$

La probabilidad de regresar al estado E_j es:

$$f_j = \sum_{n=1}^{\infty} f_j^n$$

Entonces cuando $f_j = 1$ se garantiza que se volverá a pasar por E_j y a dicho estado se le llamará recurrente.

Estado transitorio: es todo estado E_j tal que $f_j < 1$, es decir, no hay garantías de volver a dicho estado.

Estado ergódico: Son los recurrentes, aperiódicos y no nulos. Son de vital importancia para clasificar cadenas de Markov y para probar la existencia de distribuciones de probabilidad límite.

2.1.5 Clasificación de cadenas

En esta sección veremos los tipos de cadenas que nos podemos encontrar:

Cadena irreducible: Todo estado se puede alcanzar desde cualquier otro en un número finito de pasos. Es decir, para todo $i, j = 1, 2, \ldots, m$ se tiene que $p_{ij}^{(n)} > 0$ con $n \in \mathbb{N}$.

Una propiedad importante de este tipo de cadenas es que todos los estados son del mismo tipo y tienen el mismo periodo. Como consecuencia de esto, si conocemos un estado E_j de una cadena irreducible, entonces los conocemos todos.

Conjunto cerrado en una cadena de Markov: Sea $C = \cup_{j=r}^{j=s} E_j$ $1 \leq r \leq s \leq m$ un conjunto de estados en una cadena de Markov, decimos que C es cerrado si cualquier estado de C solo puede alcanzarse desde un estado dentro de C. Esto es $p_{ij} = 0$ para todo $E_j \in C$ y $E_i \notin C$.

En particular, los estados absorbentes son cerrados de un solo elemento.

Cadena ergódica: Es aquella cadena en la que todos sus estados son ergóticos.

Cadenas periódicas: Si E_j es periódico, entonces todo estado E_i tal que $p_{ij} > 0$ es periódico con el mismo periodo. Así, el periodo es una característica común del conjunto irreducible y cerrado, luego se puede hablar del periodo de una subcadena irreducible.

2.2 Modelos propuestos

En esta sección veremos dos modelos, hemos desarrollado ideas leyendo el libro [6]. Uno con 4 estados cuyas probabilidades serán determinadas en función de los resultados del capítulo anterior, y otro con 6 estados algo más elaborado con la desventaja de que será más complicado hallar la matriz de transición, de hecho esta tarea se dejará como trabajos futuros.

2.2.1 Cadena de los 6 estados

Estados

Podemos modelar un partido de fútbol como una cadena de Markov con los siguientes estados:

E_1= posesión local defensiva

E_2= posesión local ofensiva

E_3= gol local

E_4= Posesión visitante defensiva

E_5= posesión visitante ofensiva

E_6= gol visitante

Como podemos apreciar, en esta cadena estamos observando estados en los que el balón está en juego. Vamos a asumir que de los 90 minutos que tiene un partido se juegan una media de 25 por parte, a su vez supondremos dos cambios de estado por minuto. Luego nuestra cadena de Markov tendrá 200 movimientos.

Probabilidades de transición

Vamos a determinar las p_{ij}. Se tiene que $0 \leq p_{ij} \leq 1$.

Vamos a mostrar entre qué estados tenemos que $p_{ij} = 0$.

Defensa local (E_1) y defensa visitante (E_4)

Como la probabilidad de hacer gol desde el campo propio para cualquier equipo es muy pequeña, tanto que es despreciable podemos asumir que $p_{13} = p_{46} = 0$.

También se tiene que si un equipo tiene la pelota en su campo, la probabilidad de que se haga autogol es nula, esto es $p_{16} = p_{43} = 0$.

Ataque local (E_2) y ataque visitante (E_5)

Un ataque no puede acabar en ataque del equipo contrario sin ningún estado de por medio, luego afirmamos que $p_{25} = p_{52} = 0$.

Es obvio que un ataque de un equipo no puede desencadenar en gol del contrario directamente, esto es $p_{26} = p_{53} = 0$.

Gol local (E_3) y gol visitante (E_6)

Un gol implica saque de centro para el contrario, consideraremos este un balón en defensa para el equipo que ha recibido el gol. Por tanto $p_{34} = p_{61} = 1$. Como consecuencia de esto

$$p_{31} = p_{32} = p_{33} = p_{35} = p_{36} = 0$$

$$p_{62} = p_{63} = p_{64} = p_{65} = p_{66} = 0$$

Con lo cual la matriz estocástica de esta cadena es la siguiente:

$$T^{6\times 6} = \begin{pmatrix} p_{11} & p_{12} & 0 & p_{14} & p_{15} & 0 \\ p_{21} & p_{22} & p_{23} & p_{24} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ p_{41} & p_{42} & 0 & p_{44} & p_{45} & 0 \\ p_{51} & 0 & 0 & p_{54} & p_{55} & p_{56} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Tipos de estados y subcadenas

En esta cadena no existen los estados absorbentes, ya que los goles de un equipo implican necesariamente balón en defensa para el contrario, y además nadie tiene garantías de quedarse el balón hasta el final del partido.

Todos los estados son periódicos ya que en ningún momento ocurre que $p_{ii}^n = 0$. Dicho en términos menos formales, mientras se esté jugando un partido de fútbol no se puede garantizar que un equipo no vuelva a defender ni atacar, así como hacer gol.

Todos los estados son recurrentes, es sencillo explicar el caso de las posesiones de balón (1,2,4 y 5) ya que en pocos intentos se van repitiendo, sin embargo para los goles (3 y 6) no se garantiza que ningún equipo marque o vuelva a marcar, sin embargo si hacemos el partido tender a infinito, tal como podemos ver en la definición de sucesión recurrente enunciada en el apartado anterior, se tiene que aunque la probabilidad sea pequeña si haces infinitos intentos hay probabilidad 1 de que ocurra. Esta explicación también es válida para decir que todos los estados son transitorios. Sin embargo, un partido de fútbol es largo y aunque no hayan garantías a probabilidad 1, podemos afirmar que se van a seguir sucediendo al menos los estados de posesión.

Por último, al ser todos los estados periódicos no da la posibilidad a la existencia de estados ergódicos.

En cuanto a la cadena, es irreducible ya que se puede llegar de todo estado a todo estado en un número finito de pasos. Se observan también conjuntos cerrados, por ejemplo los goles (3 y 6) son cerrados, ya que solo se pueden alcanzar desde el ataque del equipo correspondiente en este modelo ($E_2 \to E_3$ y $E_5 \to E_6$).

Programación de la cadena de Markov

Vamos a hacer una función en R que simule el número de cadenas que deseemos con un número n de pasos. Concretamente, para cada partido generaremos una cadena para cada parte. En una de ellas el estado inicial será E_1 (saca de centro el local) y en la otra E_4 (saca de centro el visitante). Lo haremos con las siguientes funciones:

En primer lugar definimos una función que nos simula una cadena de Markov:

```
simula_cadena <- function(n,X0,P){
  dim <- length(P[1,])
```

```
  Xn <- numeric((n+1))
  Xn[1] <- X0
  for(i in 2:(n+1)){
    aux <- Xn[i-1]
    Xn[i] <- sample(1:dim,1,T,P[aux,])
  }
  Xn
}
```

Donde el parámetro n es el número de cambios de estado que tiene la cadena, X_0 el estado inicial y P la matriz estocástica.

Primero definimos *dim* como la longitud de las filas de la matriz P, que en nuestro caso serán 6 estados, esto lo hacemos con la orden $dim < -length(P[1,])$. A continuación definimos un vector en el que iremos guardando la suceción de estados de la simulación, a este le llamaremos X_0, que tendrá $n+1$ componentes puesto que hay n cambios de estado más el estado inicial. Esto lo definimos con $Xn < -numeric((n+1))$, donde predefinimos que X_0 será el primer estado. Esto es $Xn[1] < -X_0$. Para el resto de estados usamos el comando *sample* que continúa con la simulación y guarda en cada componente del vector Xn (a partir de la segunda componente) el resultado de la simulación "cambio de estado".

En segundo lugar crearemos otra función, que a partir de la simulación anterior obtenga el resultado de goles y también de posesión del balón:

```
estados_a_datos <- function(partido){
  tabla <- table(partido)
  c(tabla[3], tabla[6], 100*sum(tabla[1:2])/sum(tabla[-c(3,6)]))
}
```

Se obtiene una tabla que contenga el número de veces que se han dado los estados $i = 1, 2, 3, 4, 5, 6$. Esto lo hacemos con la orden $tabla < -table(partido)$. Ahora lo ideal sería obtener el resultado del partido y la posesión del balón, lo haremos de la siguiente forma:

Los goles del equipo local es la tercera componente de la tabla es decir $tabla[3]$. Análogamente los goles del visitante es similar pero con la sexta componente de la tabla, esto es $tabla[6]$.

Por otro lado, la posesión del balón se interpreta como el número de estados en que el equipo indicado tiene el balón, para el local sería el número de veces que se dan los estados 1 y 2 dividido entre el número de cambios de estado exceptuando los goles, que ahí nadie tiene la posesión, es decir $100 * sum(tabla[1:2])/sum(tabla[-c(3,6)])$ (nótese que multiplicamos por 100

para tener la posesión en porcentaje). Para el equipo visitante sería un proceso totalmente análogo con la salvedad de que en lugar de usar los estados 1 y 2, usaremos el 4 y 5, o de otro modo, Posesión visitante = 100-posesión local. Como la posesión es complementaria, solo será necesario calcular la de un equipo, lo haremos con el local.

Por último para proceder a las simulaciones lo haremos de la siguiente manera, definiendo nosotros mismos unas probabilidades de transición que nos sirvan para el ejemplo:

```
p0 <- c(.01, .33 , 0, .33, .33, 0)
p1 <- c(.33, .33, .01, .33, .33, 0)
p2 <- c( 0,  0,  0,  1,  0, 0)
p3 <- c(.33, .01, 0, .33, .33, 0)
p4 <- c(.33, 0, 0, .33, .33, .01)
p5 <- c(1, 0, 0, 0, 0, 0)
P <- rbind(p0,p1,p2,p3,p4,p5)


n <- 100
stats_simuladas <- matrix(0,n,3)
for (i in 1:n) {
  partido <- factor(c(simula_cadena(250,1,P),

   simula_cadena(255,4,P)), levels=1:6)
  stats_simuladas[i,] <- estados_a_datos(partido)
}
```

Introducimos la matriz estocástica definida anteriormente y tomaremos 100 simulaciones del mismo partido ($n < -100$). Guardaremos nuestros goles y posesión simulados como $stats_simuladas$ en una matriz de $n = 100$ filas y 3 columnas (goles local, goles visitante y posesión local). A continuación con el for procedemos a hacer las 100 simulaciones del mismo partido con las probabilidades de la matriz estocástica P distinguiendo la primera de la segunda parte del partido. Esto es:

```
partido <- factor(c(simula_cadena(250,1,P),

simula_cadena(255,4,P)), levels=1:6)
```

Finalmente pasamos el vector de estados de cada partido a un vector de tres componentes que serán el resultado y la posesión.

```
stats_simuladas[i,] <- estados_a_datos(partido)
```

Ejemplo 2.3.
```
> partido
  [1] 1 2 2 2 4 1 2 5 4 1 2 4 1 4 4 1 4 4 1 2 4 5 1 4 1 4 1 ...
 [51] 5 4 1 1 4 4 5 5 4 5 1 2 5 1 4 4 4 1 2 2 5 5 5 4 4 1 4 ...
[101] 2 4 2 4 5 4 5 1 2 1 5 5 5 4 1 5 1 4 5 1 5 5 4 5 5 5 5 ...
[151] 4 1 5 5 4 4 1 4 1 5 1 4 5 1 4 4 5 1 5 1 5 5 1 2 5 4 4 ...
[201] 4 5 4 5 1 4 5 5 5 1 4 1 5 4 4 4 4 4 5 1 5 4 1 2 4 1 2 ...
[251] 5 4 4 5 5 1 4 4 1 4 4 4 1 2 1 2 4 1 4 1 5 5 4 5 1 5 1 ...
[301] 5 1 5 4 4 4 1 2 4 5 4 5 4 4 1 2 5 5 1 4 5 4 4 4 1 5 1 ...
[351] 5 1 2 2 2 5 4 1 4 5 1 2 4 5 5 4 5 4 1 4 4 1 4 1 4 4 4 ...
[401] 4 5 4 4 5 5 5 5 5 4 5 4 5 1 2 1 2 5 4 5 4 4 4 5 5 1 2 ...
[451] 4 4 4 5 4 4 1 2 4 1 2 1 5 1 2 1 4 4 1 2 4 4 4 1 5 5 1 ...
[501] 4 5 1 4 1 4 1
Levels: 1 2 3 4 5 6
```

Esto es un ejemplo de simulación de un partidos y la sucesión de estados que se ha ido dando. Con la siguiente orden no solo veremos el número de veces que se da el estado 3 y el 6, esto es, el marcador, sino que con la suma de los estados 1 y 2, 3 y 4 respectivamente podremos conocer la posesión del balón local y visitante, en nuestro caso solo mostraremos la del local ya que la otra es el complementario respecto de 1. Haremos esto para varias simulaciones del mismo partido.

```
> stats_simuladas
      [,1] [,2]     [,3]   [,1]  [,2]     [,3]
 [1,]    0    0 37.67258   [2,]    0    0 34.91124
 [3,]    1    2 37.10317   [4,]    0    0 34.31953
 [5,]    0    0 40.23669   [6,]    0    1 38.53755
 [7,]    0    5 38.64542   [8,]    0    2 35.44554
 [9,]    0    5 35.85657  [10,]    1    5 37.32535
[11,]    0    3 30.75397 [12,]    0    7 37.60000
[13,]    0    1 34.78261 [14,]    1    0 36.16601
[15,]    0    1 38.73518 [16,]    0    1 33.00395
[17,]    0    1 39.52569 [18,]    0    2 35.44554
[19,]    1    5 33.33333 [20,]    0    1 34.78261
[21,]    1    0 35.17787 [22,]    1    0 34.38735
[23,]    0    4 33.20080 [24,]    0    2 38.61386
[25,]    0    2 35.04950 [26,]    1    2 35.31746
[27,]    1    2 36.70635 [28,]    0    4 34.99006
[29,]    0    1 34.18972 [30,]    0    1 35.37549
[31,]    1    2 37.50000 [32,]    0    1 35.17787
[33,]    0    1 33.00395 [34,]    1    2 37.89683
[35,]    0    0 36.8836  [36,]    0    2 33.06931
[37,]    3    2 32.07171 [38,]    0    2 32.47525
[39,]    0    0 35.89744 [40,]    1    1 38.21782
```

```
[41,]    1    2 32.53968 [42,]    0    1 34.58498
[43,]    0    2 36.43564 [44,]    0    1 31.42292
[45,]    0    2 35.44554 [46,]    0    1 34.58498
[47,]    0    2 32.47525 [48,]    1    1 33.06931
[49,]    0    3 33.73016 [50,]    0    2 38.21782
[51,]    0    0 36.68639 [52,]    0    1 36.36364
[53,]    0    3 38.49206 [54,]    0    2 39.00990
[55,]    0    1 37.94466 [56,]    0    5 41.43426
[57,]    0    1 34.58498 [58,]    0    1 35.57312
[59,]    1    0 35.77075 [60,]    1    4 34.06375
[61,]    1    1 34.85149 [62,]    1    3 35.58648
[63,]    1    3 35.98410 [64,]    1    1 38.21782
[65,]    0    0 33.33333 [66,]    1    2 34.32540
[67,]    0    0 35.10848 [68,]    0    3 41.66667
[69,]    1    2 36.90476 [70,]    0    0 39.84221
[71,]    0    3 40.47619 [72,]    0    1 34.38735
[73,]    0    1 30.23715 [74,]    1    2 39.48413
[75,]    1    1 36.43564 [76,]    0    2 34.05941
[77,]    0    2 33.06931 [78,]    1    1 39.80198
[79,]    0    0 34.51677 [80,]    0    1 31.22530
[81,]    0    1 33.99209 [82,]    0    1 33.59684
[83,]    0    2 36.23762 [84,]    0    2 32.47525
[85,]    0    1 39.72332 [86,]    0    0 38.65878
[87,]    1    1 38.61386 [88,]    0    1 34.58498
[89,]    0    1 35.96838 [90,]    1    2 36.11111
[91,]    0    4 36.97813 [92,]    0    3 39.28571
[93,]    1    2 34.32540 [94,]    0    2 32.87129
[95,]    0    2 34.65347 [96,]    1    4 35.25896
[97,]    1    1 35.64356 [98,]    0    1 35.37549
[99,]    1    1 40.99010[100,]    0    0 37.47535
```

Como conclusión se observa que esta cadena teóricamente es muy factible ya que se adapta a las situaciones de un partido. El problema reside en cómo estimar las p_{ij} ya que no es una tarea fácil. Podría asumirse cada p_{ij} como una Poisson univariante siendo λ la media de veces que ocurre dicho cambio de estado en un número considerable de partidos anteriores, sin embargo tal como está definida la cadena, no son sucesos fáciles de identificar. Este modelo lo dejaremos propuesto y la determinación de las p_{ij} queda para trabajos futuros. Elaboraremos en las próxima sección otra cadena de Markov, en las que trabajaremos con partidos Reales.

2.2.2 Cadena de los 4 estados

En este nuevo modelo crearemos una cadena de Markov más sencilla con 4 estados.

estados

E_1=Balón local.

E_2=Balón visitante.

E_3=Gol local

E_4=Gol visitante

Probabilidades de transición

Vamos a determinar las p_{ij}. Se tiene que $0 \leq p_{ij} \leq 1$.

Vamos a mostrar entre qué estados tenemos que $p_{ij} = 0$. Se tiene que no se puede pasar de gol local a gol visitante, entonces $p_{34} = p_{43} = 0$, así como tampoco puede haber dos goles seguidos de un mismo equipo ya que cuando hay un gol se cambia de posesión por el saque central del contrario, luego $p_{33} = p_{44} = 0$ y $p_{32} = p_{41} = 1$. Como consecuencia de esto $p_{3i} = p_{4j} = 0$ $i \neq 2$ $j \neq 1$ También se tiene que si un equipo tiene el balón, supondremos que no se hará autogol. Esto es $p_{14} = p_{23} = 0$. Por último se asume que cada estado de posesión dura hasta que el equipo marque o pierda el balón por largo que sea, así $p_{ii} = 0$ para todo $i = 1, 2, 3, 4$. Luego solo quedan 4 probabilidades de transición por definir, que son p_{12}, p_{13}, p_{21}, p_{24}.

Nos basaremos en el modelo introductorio del capítulo anterior, asumiremos n ataques para el equipo local y m para el equipo visitante. De este modo definiremos $p_{13} = \frac{\lambda_l}{n}$ $p_{24} = \frac{\lambda_v}{m}$. Donde los λ son la media de goles de la muestra de partidos anteriores y n,m es la media de ocasiones por partido del equipo local y visitante respectivamente. Para definir las otras dos, utilizo que la suma de las filas de la matriz de transición ha de ser 1. Así, $p_{13} = 1 - p_{12}$ y $p_{21} = 1 - p_{24}$. La matriz finalmente queda de la siguiente manera:

$$T^{4\times 4} = \begin{pmatrix} 0 & 1-\lambda_l & \lambda_l & 0 \\ 1-\lambda_v & 0 & 0 & \lambda_v \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Programación de la cadena de Markov

Es análoga al caso anterior, salvo que cambia la matriz de transición, que en esta ocasión es cuadrada de orden 4. Sus probabilidades asociadadas las hemos definido en el apartado anterior. Queda codificado de la siguiente manera el fichero de R:

```
simula_cadena <- function(n,X0,P){
  dim <- length(P[1,])
  Xn <- numeric((n+1))
  Xn[1] <- X0
  for(i in 2:(n+1)){
    aux <- Xn[i-1]
    Xn[i] <- sample(1:dim,1,T,P[aux,])
  }
  Xn
}


estados_a_datos <- function(partido){
  tabla <- table(partido)
  c(tabla[3], tabla[4], 100*sum(tabla[1])/sum(tabla[-c(3,4)]))
}


p0 <- c(0, .09 , .01, 0)
p1 <- c(.09, 0, 0, .01)
p2 <- c( 0,  1,  0,  0)
p3 <- c(1, 0, 0, 0)
P <- rbind(p0,p1,p2,p3)


n <- 100
stats_simuladas <- matrix(0,n,3)
for (i in 1:n) {
  partido <- factor(c(simula_cadena(6,1,P), simula_cadena(6,2,P)),
   levels=1:4)
  stats_simuladas[i,] <- estados_a_datos(partido)
}
```

Es similar al código anterior, adaptándose para el nuevo modelo. Hemos vuelto a poner probabilidades arbitrarias para poner un código genérico. Sin embargo

ya podemos aplicarlo a partidos reales de LaLiga y La Liga Iberdrola. Vamos a mostrar un ejemplo de LaLiga, para ello necesitamos los valores de λ ya calculados para la jornada 20. Además de ello, necesitamos los valores de m y n, partiendo de que asumimos que en un partido de fútbol hay de media 100 acciones, podemos asumir que la posesión local aproximada a un número natural m y análogamente con n y la posesión visitante, ya que precisamente esta se mide entre 0 y 100. Con esta fórmula calcularemos los p_{13} y p_{24} (en consecuencia p_{12} y p_{21}). Los mostraremos en la siguiente tabla:

Equipo	posesión media	Equipo	posesión media
Barcelona	64	Valencia	47
At. Madrid	49	Levante	45
Sevilla	51	Athletic	50
R.Madrid	61	Valladolid	47
Alavés	41	Leganés	41
Getafe	39	Eibar	55
Betis	63	Celta	53
R. Sociedad	51	Rayo	46
Girona	46	Villareal	47
Espanyol	49	Huesca	43

Tabla 2.1. media de posesión por equipos.

Con esto podemos proceder con el modelo:

Ejemplo 2.4. Simular el partido de la jornada 20 entre el Eibar y el Espanyol. (Eibar local)

Primeramente calculamos p_{13} y p_{24}. La media de goles del Eibar y Espanyol de 1.105 y 1.105 respectivamente (han hecho el mismo número de goles en las 19 jornadas ligueras anteriores) en cuanto a posesión tienen una media de 55 y 49 respectivamente (sobre cien), entonces se tiene que $p_{13} = \frac{1.105}{55} = 0.0200909091$ y $p_{24} = \frac{1.105}{49} = 0.022510204$, en consecuencia, $p_{12} = 1 - p_{13} = 0.97990909$ y $p_{21} = 1 - p_{24} = 0.9774489796$.

Podemos usar el código R confeccionado para este tipo de cadenas, esta es la matriz a introducir:

```
p0 <- c(0, .98 , .02, 0)
p1 <- c(.98, 0, 0, .02)
p2 <- c( 0,  1,  0,  0)
p3 <- c(1, 0, 0, 0)
P <- rbind(p0,p1,p2,p3)
```

Véase que son dos equipos igualados. Vamos a mostrar la simulación de un partido.

```
[1] 1 2 1 3 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 ...
 [51] 2 2 1 2 1 2 1 2 4 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 ...
[101] 2 1
Levels: 1 2 3 4
```

Se observa que hay un 3 y un 4 en toda la secuencia (en lo que no hemos mostrado no hay ninguno), vamos a lanzar ahora 100 simulaciones pero solo mostrando el resultado y la posesión:

```
> stats_simuladas
      [,1] [,2]     [,3]    [,1] [,2]     [,3]
 [1,]    1    0 50.49505 [2,]    1    2 49.49495
 [3,]    0    2 50.00000 [4,]    0    0 50.00000
 [5,]    0    3 50.50505 [6,]    1    0 49.50495
 [7,]    1    2 49.49495 [8,]    5    1 50.00000
 [9,]    0    1 49.50495[10,]    0    1 50.49505
[11,]    1    3 50.00000[12,]    1    2 50.50505
[13,]    0    0 50.0000 [14,]    2    1 50.50505
[15,]    3    2 49.4845 [16,]    2    1 49.49495
[17,]    1    0 50.4950 [18,]    1    0 49.50495
[19,]    1    2 49.4949 [20,]    0    1 50.49505
[21,]    1    1 50.0000 [22,]    0    0 50.00000
[23,]    1    1 50.00000[24,]    1    1 50.00000
[25,]    0    2 50.0000 [26,]    0    2 50.00000
[27,]    0    1 49.5049 [28,]    0    1 49.50495
[29,]    3    0 49.4949 [30,]    1    4 50.51546
[31,]    1    0 49.5049 [32,]    1    0 49.50495
[33,]    1    2 49.4949 [34,]    2    2 50.00000
[35,]    1    0 50.4950 [36,]    2    1 49.49495
[37,]    0    0 50.0000 [38,]    3    1 50.00000
[39,]    0    2 50.0000 [40,]    1    0 49.50495
[41,]    1    1 50.0000 [42,]    2    0 50.00000
[43,]    1    5 50.0000 [44,]    1    1 50.00000
[45,]    1    0 49.5049 [46,]    0    1 49.50495
[47,]    2    1 49.4949 [48,]    0    1 50.49505
[49,]    0    0 50.0000 [50,]    0    1 50.49505
[51,]    3    1 50.0000 [52,]    1    1 50.00000
[53,]    0    0 50.0000 [54,]    1    0 50.49505
[55,]    1    1 50.0000 [56,]    1    1 50.00000
[57,]    0    0 50.0000 [58,]    2    0 50.00000
[59,]    0    2 50.0000 [60,]    1    1 50.00000
```

```
[61,]    0    0 50.0000 [62,]    2    0 50.00000
[63,]    1    0 50.4950 [64,]    0    1 49.50495
[65,]    1    1 50.0000 [66,]    2    1 50.50505
[67,]    0    0 50.0000 [68,]    0    1 49.50495
[69,]    2    0 50.0000 [70,]    0    2 50.00000
[71,]    1    2 49.4949 [72,]    1    2 49.49495
[73,]    2    0 50.0000 [74,]    1    3 50.00000
[75,]    0    1 49.5049 [76,]    1    0 49.50495
[77,]    1    1 50.0000 [78,]    1    0 50.49505
[79,]    1    0 50.4950 [80,]    1    1 50.00000
[81,]    1    0 50.4950 [82,]    1    0 49.50495
[83,]    0    0 50.0000 [84,]    0    0 50.00000
[85,]    1    2 49.4949 [86,]    0    1 49.50495
[87,]    0    0 50.0000 [88,]    1    3 50.00000
[89,]    0    1 50.4950 [90,]    1    0 49.50495
[91,]    1    1 50.0000 [92,]    0    1 50.49505
[93,]    1    0 49.5049 [94,]    0    1 49.50495
[95,]    0    1 49.5049 [96,]    2    1 49.49495
[97,]    1    0 49.5049 [98,]    1    2 49.49495
[99,]    1    0 50.4950[100,]    1    1 50.00000
```

Se han obtenido los siguientes resultados:

Resultado	Número de veces	Resultado	Número de veces
0-0	14	1-0	18
2-0	5	3-0	1
0-1	14	0-2	6
0-3	1	1-1	15
2-1	6	3-1	2
5-1	1	1-2	10
1-3	3	1-5	1
2-2	2	3-2	1

Tabla 2.2. número de veces de cada resultado en la simulación

References

[1] Karlis D. Ntzoufras I. (2005) Bivariate Poisson and Diagonal Inflated Bivariate Poisson Regression Models in R. *Journal of Statistical Software.* 14(10) doi:10.18637/jss.v014.i10

[2] Karlis D. Ntzoufras I. (2007). bivpois: Bivariate Poisson Models Using The EM Algorithm. R package version 0.50-3. http://www.stat-athens.aueb.gr/ jbn/papers/paper14.htm

[3] R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[4] Rocca, J. (2019) A brief introduction to Markov Chains. Definitions, propieties and PageRank example. *Towards data Science.* https://towardsdatascience.com/brief-introduction-to-markov-chains-2c8cab9c98ab

[5] Ross. S. (2002) *A First Course of Probability.* Sixth edition, Prentice Hall.

[6] Sadovskii L.E. Sadovskii A.L. (1994) *Mathematics and Sports.* American Mathematical Society.

[7] Teutonico D. (2013) Instant R Starter. Packt Publishing.

Probabilistic models for the estimation of sport score

Juan Diego Fernández García
Facultad de Ciencias · Sección de Matemáticas
Universidad de La Laguna
alu0100776324@ull.edu.es

Abstract

This work will present two probabilistic models whose main tool is the distribution of Poisson. The aim of these two proposals is to assign each football match a probability distribution for the possible outcomes. In the first proposal we will assume independence among the goals that will mark the opposing teams. The alternative model is based on the bivariate Poisson, which assumes correlation between the number of goals of both teams. Both methods are analyzed and compared using data from LaLiga and La Liga Iberdrola. We will also propose a party model as a Markov chain that allows us to do simulations. In particular in this report, examples with data from LaLiga and La Liga Iberdrola 2018-19 will be included.

1. Introduction

Football is the world is leading sport, so it is the sport that we are going to discuss in this project, both in the male and female categories. In particular we will analyze two samples of data that comprise the first round of LaLiga and La Liga Iberdrola, the two most important competitions in Spain in men's and women's football respectively.
To conclude the work we will define Markov chains, which based on the probabilities calculated in Chapter 1, being more or less precise, we can simulate each LaLiga game and the Iberdrola League with the data that we have. In addition we will leave the computational algorithm so that taking the odds with any other method can also be used and so do the relevant game simulations.

2. The Poisson distribution

In this section we will define the distribution of Poisson, whose probability function is as follows:

$$Pois(x) = P(X = x) = \frac{e^{-\lambda}\lambda^x}{x!}$$

For the prediction model we will take the λ for each match as the arithmetic mean of the goals of the local team and analogously with the visitor, the fact that a team is local or visiting can be avoided or a local and visiting average can be distinguished for each team.
We will also define the Poisson bivariate, which has the probability function $f_{BP}(x, y) = P(X = x, Y = y)$ where:

$$P(X = x, Y = y) = e^{-(\lambda_1+\lambda_2+\lambda_3)}\frac{\lambda_1^x}{x!}\frac{\lambda_2^y}{y!}\sum_{k=0}^{min(x,y)}\binom{x}{k}\binom{y}{k}k!(\frac{\lambda_3}{\lambda_1\lambda_2})^k$$

To predict the λ of this model we use the regression model of Karlis and Ntzoufras (2003), the EM Algorithm (Expectation - Maximization) will be applied.
With the λ calculated, we can consider our Poisson bivariant as the multiplication of two Poissons independent of the form:

$$BP(\lambda_1, \lambda_2, \lambda_3) = Pois(\lambda_1 + \lambda_3) + Pois(\lambda_2 + \lambda_3)$$

We will do a study to see the quality of the models in terms of the average amount of hits per day in each sample.

3. Markov Chains

In this chapter we will introduce the concept of Markov chain, a probabilistic method with a series of events, which have an assigned probability. They are formed by events, which we will call states, and transitions that are state changes. If we form a matrix with all P_{ij} status changes, this is defined as a stochastic matrix. In the attached memory there is a theoretical framework where we will define the main notions on this topic, as well as properties, state types and chain types. We will propose two chains for football matches, one of them with 6 states with the following transition matrix:

$$T^{6\times6} = \begin{pmatrix} P_{11} & P_{12} & 0 & P_{14} & P_{15} & 0 \\ P_{21} & P_{22} & P_{23} & P_{24} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ P_{41} & P_{42} & 0 & P_{44} & P_{45} & 0 \\ P_{51} & 0 & 0 & P_{54} & P_{55} & P_{56} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

For definition reasons it is a difficult task to assign P_{ij} not null to this stochastic matrix. Define another string of 4 states in which based on the independent Poisson model we can do it, its transition matrix has this form:

$$T^{4\times4} = \begin{pmatrix} 0 & 1-\lambda_l & \lambda_l & 0 \\ 1-\lambda_v & 0 & 0 & \lambda_v \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

References

[1] Karlis D. Ntzoufras I. (2005) Bivariate Poisson and Diagonal Inflated Bivariate Poisson Regression Models in R. *Journal of Statistical Software*. 14(10) doi:10.18637/jss.v014.i10

[2] Karlis D. Ntzoufras I. (2007). bivpois: Bivariate Poisson Models Using The EM Algorithm. R package version 0.50-3. http://www.stat-athens.aueb.gr/ jbn/papers/paper14.htm

[3] R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[4] Rocca, J. (2019) A brief introduction to Markov Chains. Definitions, propieties and PageRank example. *Towards data Science.* https://towardsdatascience.com/brief-introduction-to-markov-chains-2c8cab9c98ab

[5] Ross. S. (2002) *A First Course of Probability*. Sixth edition, Prentice Hall.

[6] Sadovskii L.E. Sadovskii A.L. (1994) *Mathematics and Sports*. American Mathematical Society.

[7] Teutonico D. (2013) Instant R Starter. Packt Publishing.