

TÉCNICAS DE MINERÍA DE DATOS APLICADAS EN BALONCESTO



UNIVERSIDAD DE SEVILLA

FACULTAD DE MATEMÁTICAS

**DEPARTAMENTO DE ESTADÍSTICA E INVESTIGACIÓN
OPERATIVA**

DOBLE GRADO EN MATEMÁTICAS Y ESTADÍSTICA

Juan Antonio Velázquez Carricondo

©2018

TÉCNICAS DE MINERÍA DE DATOS
APLICADAS EN BALONCESTO

Por

Juan Antonio Velázquez Carricondo

TRABAJO FIN DE GRADO
PRESENTADO PARA OPTAR
AL DOBLE GRADO EN MATEMÁTICAS Y ESTADÍSTICA
POR LA UNIVERSIDAD DE SEVILLA

Juan Antonio Velázquez Carricondo

UNIVERSIDAD DE SEVILLA
DEPARTAMENTO DE
ESTADÍSTICA E INVESTIGACIÓN OPERATIVA

El abajo firmante ha leído este trabajo y recomienda a la Facultad de Matemáticas la aceptación del trabajo “**Técnicas de Minería de Datos Aplicadas en Baloncesto**” realizado por **Juan Antonio Velázquez Carricondo** como Trabajo Fin de Grado para obtener el Doble Grado en Matemáticas y Estadística conforme a lo dispuesto en la Ley.

Con fecha de: Junio 2018

Tutor:

Prof. Dr. José María Fernández Ponce

UNIVERSIDAD DE SEVILLA

Fecha: **Junio 2018**

Autor: **Juan Antonio Velázquez Carricondo**

Título: **Técnicas de Minería de Datos Aplicadas en
Baloncesto**

Dpto: **Estadística e Investigación Operativa**

Firma del autor

SE PROHIBE LA REPRODUCCIÓN TOTAL O PARCIAL DE ESTE
TRABAJO SIN EL PERMISO EXPRESO DEL AUTOR O DE LA UNIVERSIDAD
DE SEVILLA.

Resumen	I
Abstract	III
1. INTRODUCCION	1
1.1. Introducción	1
1.1.1. Historia	2
1.1.2. Objetivos	3
1.1.3. Contenido	4
2. ANALISIS DE REDES	7
2.1. Redes Neuronales Artificiales	7
2.2. Introducción	7
2.3. ¿Qué es una Red de Neuronas Artificiales?	8
2.3.1. Introducción biológica	8
2.3.2. Elementos de una Neurona Artificial	9
2.3.3. Redes neuronales	13
2.4. El Perceptrón Simple	15
2.5. El Perceptrón Multicapa	18
2.6. <i>Deep Learning</i>	22
2.6.1. Introducción	22
2.6.2. Optimalidad global en <i>DL</i>	24
2.6.3. El reto de las funciones no convexas en las redes neuronales	24
2.6.4. Estabilidad geométrica en el <i>DL</i>	25
2.7. Aplicaciones al baloncesto	27
2.7.1. Objetivos	27
2.7.2. Datos	28
2.7.3. Tratamiento de los datos	29
2.7.4. Aplicaciones	30
2.7.5. Conclusiones	33
3. ANÁLISIS ESPACIAL	37
3.1. Introducción	37
3.2. Resultados teóricos	37
3.2.1. Procesos estocásticos	38

3.2.2.	Procesos puntuales espaciales	38
3.2.3.	Procesos Gaussianos	38
3.2.4.	Procesos de Poisson	39
3.2.5.	Procesos Log-Gaussianos Cox	42
3.2.6.	Matriz de Factorización No Negativa	42
3.3.	Datos	44
3.3.1.	Observación de la No Estacionariedad	45
3.4.	Idea aproximada	46
3.4.1.	Ajuste de LGCPs	47
3.4.2.	Optimización NMF	48
3.5.	Resultados	49
3.6.	Eficiencia del tiro	50
3.6.1.	Descripción del modelo	52
3.6.2.	Inferencia	54
3.6.3.	Resultados	54
3.7.	Conclusiones	56
4.	MÉTRICAS	59
4.1.	Introducción	59
4.1.1.	Conceptos previos	59
4.1.2.	Rating Ofensivo	63
4.1.3.	Teoría de la credibilidad	64
4.2.	Método	65
4.3.	Resultados	68
4.3.1.	Hiperparámetros	69
4.4.	Discusión	70

En este trabajo, se pretende dar una visión general de las distintas aplicaciones del *Machine Learning* al deporte y en particular al baloncesto. Se presentan tres bloques muy distintos entre sí para concienciar sobre todas las posibilidades que ofrece la estadística y el análisis de datos al ámbito deportivo. En el primero se introduce el concepto de Redes Neuronales, el Perceptrón Simple, el Perceptrón Multicapa y se explica brevemente el funcionamiento del *Deep Learning*, aplicándose a un modelo que predice la posición en la que juegan los distintos jugadores en función de distintas variables. El segundo bloque se centra en el estudio espacial de los hábitos de tiro de los distintos jugadores y de la eficiencia que tienen los mismos en las distintas zonas de la pista, en ambos casos se aplicará conceptos de Inferencia Bayesiana. Por último, se compararán distintas métricas ofensivas y se introducirán nuevas métricas del rendimiento ofensivo bajo un enfoque bayesiano, para así por último comparar dichas métricas y observar de qué manera reflejan el comportamiento del jugador en la pista.

In this paper, it is pretended to introduce a huge vision about how Machine Learning can be used in sports, and particularly in Basketball. This document is composed of three different parts in order to raise awareness about how much possibilities advanced statistics offers (in) sports. In the first, Neural Networks, the Simple Perceptron and the Multilayer Perceptron concepts are introduced and a brief explanation of how Deep Learning works applying it into a model, that predicts the position of the different players using some variables, is also given. The second part is focused on studying shots spacial habits in order to distinguish different types of players. After that, the efficiency of them is also studied from a spatial view of points. In both cases, Bayesian concepts have been used. At the end, some different offensive metrics are compared and a new one is developed using Bayesian methods, in order to compare them and see how much players performance are written on them.

1 INTRODUCCION

*“No puedes elegir el modo de perder,
pero si puedes elegir cómo recuperarte
para ganar la próxima vez”*

Pat Riley (1945-)

1.1. Introducción

El papel de las matemáticas tiene cada vez un rol más importante en nuestra sociedad. Son muchos los ámbitos de nuestra vida en los que está presente de una forma u otra, consultorías, bancos, empresas de transporte, compañías de vuelos, aseguradoras, equipos médicos son algunos de los ejemplos más relevantes donde su uso se torna fundamental.

Entre diversas funciones, destacan algunas como la aplicación de técnicas estadísticas para la explotación de datos de cara a asesorar la toma de decisiones, la optimización de diversos procesos, que pueden ir desde la obtención del mejor recorrido para un repartidor de Amazon a la planificación de los vuelos de compañías aéreas o la obtención de modelos estadísticos que permitan hacer una idea aproximada de cómo funciona un determinado proceso.

En el ámbito del deporte, aunque su uso va incrementándose poco a poco, hay muchos sectores del mismo que son todavía reacios a la aplicación de técnicas estadísticas para diversos aspectos deportivos, como pueden ser el análisis técnico-táctico, el análisis de rendimiento o la prevención de lesiones entre otros aspectos. En cambio, los resultados obtenidos por ciertos equipos tras la incorporación de dichas técnicas (cabe destacar equipos como *Houston Rockets* en la *NBA* con directores como Winston Lin y Fan-Hal

Koung, *Leicester City* en la *Premier League* con Andy Blake o el *FC Barcelona* con su departamento denominado “Barça Innovation Hub” en *La Liga Santander*) hacen que el uso de las matemáticas en el ámbito del deporte sea cada vez más usual. En este trabajo se pretende describir diversas técnicas estadísticas aplicadas al baloncesto para mostrar el amplio espectro de la estadística en el deporte.

1.1.1. Historia

La primera vez que se utilizó la Estadística en el ámbito deportivo fue en Béisbol, concretamente a principios de la década de los 90, cuando Billy Beane, *manager* de los *Oakland Athletics*, con ayuda de Paul DePodesta (experto en estadística y aficionado a Béisbol), se basó en estadísticas avanzadas para fichar a jugadores que fueran baratos (por supuestos defectos) y que a su vez, conjuntamente, formaran un equipo competitivo. El resultado fue inmejorable y consiguieron batir el récord de partidos ganados seguidos de la temporada regular, mostrando el valor de la Estadística. A partir de ahí su uso se ha ido incrementando poco a poco pero con paso firme y extendiéndose a otros deportes.

Hoy en día, la Estadística es fundamental en la NBA y en todos los equipos hay un departamento dedicado sólo y exclusivamente a su uso en diversos aspectos del juego, que van desde el modelado de movimientos ofensivos (véase Wu y Bornn (2018)) y aspectos tácticos del juego (véase McQueen, Wiens, y Guttag (2014)) a prevención de lesiones (véase Ward, Tankovich, Ramsden, Drust, y Bornn (2018) y Dower, Rafehi, Weber, y Mohamad (2018)) y detección de nuevos talentos (véase Johnston, Wattie, Schorer, y Baker (2018)). Ya es raro que haya equipos en cualquier deporte que no utilicen un mínimo de estadística en sus análisis, y en los próximos años, con la implementación

del *Big Data* hará que la estadística sea parte fundamental en el deporte.

1.1.2. Objetivos

En este proyecto se pretende acercar el uso de la estadística en el deporte, concretamente al baloncesto, mostrando las diversas posibilidades que tiene el uso de la estadística en el mismo. El recurso más importante para la Minería de Datos es la posesión de una base de datos que se pueda explotar para obtener toda la información posible. Aunque la mayoría relaciona la Minería de Datos en el deporte con la predicción de resultados y la utilización de herramientas para desarrollar nuevas métricas que reflejen mejor el rendimiento de los jugadores y del equipo, la Minería de Datos va más allá e incluye técnicas para ayudar a la toma de decisiones de los entrenadores y de los clubs, como pueden ser la elección de jugadores, el control de los salarios o la valoración de nuevos fichajes.

El objetivo es la introducción de técnicas de *Machine Learning* como son las *Redes Neuronales Artificiales* para diversos aspectos y posibles usos de dichas técnicas en futuros proyectos. También se mostrará distintos modelos estadísticos que estudian los hábitos de tiro de los distintos jugadores de la NBA y la efectividad de los mismos, lo que permite a los equipos analizar rápidamente a sus rivales o los posibles defectos de sus jugadores. Por último, se pretende introducir el uso de las métricas en el baloncesto para optimizar la representación de cómo es el jugador en dicho aspecto del juego.

1.1.3. Contenido

Este proyecto se dividirá en tres partes claramente diferenciadas, con el objetivo de mostrar tres aplicaciones al baloncesto muy distintas entre sí.

1. En el Capítulo 2, se explicará en qué consisten las *Redes Neuronales Artificiales*, su base teórica, el *Perceptrón Simple*, el *Perceptrón Multicapa* y también los fundamentos del *Deep Learning* (DL), que es la técnica actual más empleada dentro de redes neuronales. A modo de ejemplo, se tomará un partido jugado a jugada del repositorio de datos de GitHub Rajiv Shah (22 de Agosto de 2013) y tratando los datos correctamente para poder hacer distintos modelos con el software libre **R** aplicando técnicas de *Deep Learning*. Se pretende obtener un modelo que prediga la posición en la que juega dicho jugador en función de variables que describiremos en dicho capítulo.
2. En el Capítulo 3, se utilizarán los datos de los tiros de campo intentados y anotados con sus correspondientes localizaciones de más o menos la mitad de los partidos de la temporada regular 2012-2013 de la NBA. Con estos datos se pretende aplicar distintos modelos estadísticos, basados en los *Procesos de puntos*, *Procesos de Poisson*, *Procesos Gaussianos* y *Procesos Log-Gaussianos Cox* junto a otros métodos de reducción de la dimensión, como es *Matriz de Factorización No Negativa* para, en primer lugar realizar una descripción de los hábitos de tiro de los distintos jugadores, de cara a obtener con dichos datos una idea rápida de cómo se comporta el jugador en la pista y que características tiene en cuanto al tiro, lo cual es muy importante para plantear una defensa efectiva. En segundo lugar, una vez se haya analizado los hábitos de tiro de los distintos jugadores, se

analizará más detenidamente la eficiencia de los mismos, para realizar un análisis más profundo del tiro y a su vez para ver cuáles son los jugadores más determinantes.

3. En el Capítulo 4, se hará una introducción a algunas métricas utilizadas en el baloncesto, las cuales dan una idea general del rendimiento de los jugadores en determinados aspectos, tanto ofensivos como defensivos. Aunque en dicho capítulo nos centraremos en los índices ofensivos, en particular, se estudiarán aquellos que miden el impacto ofensivo de los distintos jugadores y se hará una comparación con las medias de una familia conjugada *Multinomial-Dirichlet* para obtener la eficiencia en la anotación de los jugadores (SEff). Lo que se pretende, es obtener un punto de vista distinto en las métricas que miden la eficiencia en la anotación de los distintos jugadores para compararlas con la métrica nueva que se introducirá en dicho capítulo, para ver qué jugadores están sobrevalorados o infravalorados y tener una idea de su impacto ofensivo de forma más detallada.

2 ANALISIS DE REDES

*“El talento gana partidos,
pero el trabajo en equipo y la inteligencia
ganan campeonatos”*

Michael Jordan (1963-)

2.1. Redes Neuronales Artificiales

2.2. Introducción

A lo largo de este capítulo se introducen las redes neuronales artificiales, método muy utilizado en estadística computacional y con un gran potencial en aplicaciones específicas como la detección de transacciones fraudulentas con tarjetas de crédito, detección de minas, frenos ABS, reconocimiento de caracteres, etc. Se verán los conceptos básicos como *Neurona* y *Redes Neuronales Artificiales*, al igual que los elementos que las componen. Una vez se haya visto los conceptos básicos, se describirá el *Perceptrón Simple* y el *Perceptrón Multicapa* (véase Larranaga, Inza, y Moujahid (2018)). Por último, se aplicará las redes neuronales artificiales a la detección de patrones tácticos en baloncesto, para ello introduciremos los métodos de *Aproximación Jerárquica* y de *Aproximación Recursiva*.

2.3. ¿Qué es una Red de Neuronas Artificiales?

2.3.1. Introducción biológica

El objetivo principal de una *red de neuronas artificiales* es emular el comportamiento, en la medida de lo posible, de una red de neuronas biológicas.

Una neurona está compuesta por:

- Soma: Cuerpo celular.
- Dendritas: Denso árbol de ramificaciones que surgen del cuerpo celular.
- Axón: Fibra tubular que parte del soma.
- El axón se ramifica en su extremos final y es de la forma en la que la neurona se conecta con otras neuronas.

Hay que observar que la información fluye desde las dendritas hacia el axón, atravesando el soma (ver Figura 2.1).

Con una red de neuronas artificiales pretendemos simular, como dijimos anteriormente, varias propiedades como:

- PROCESAMIENTO PARALELO: Pretendemos trabajar con una cantidad enorme de procesadores de información sencillos (neuronas) a la vez, para realizar trabajos mucho más complejos.

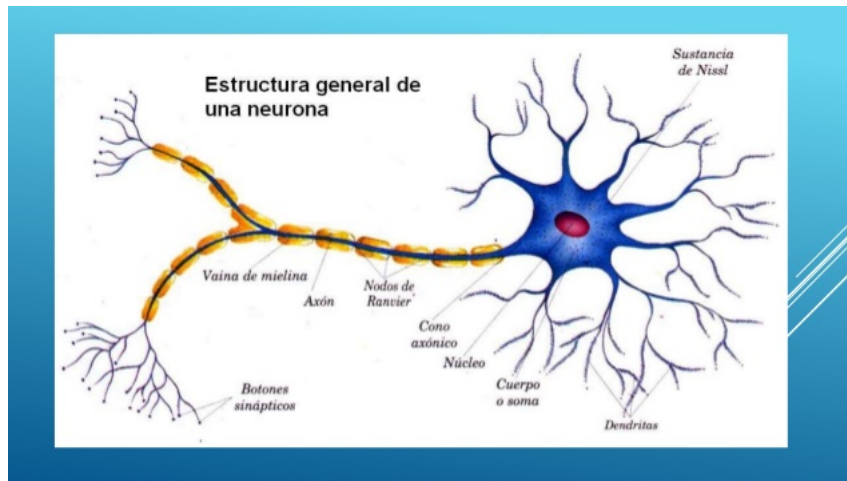


Figura 2.1: Estructura biológica de una neurona

- **MEMORIA DISTRIBUIDA:** La información está distribuida en la sinapsis de la red, habiendo una redundancia en el almacenamiento, aunque evita la pérdida de información en caso de que la red sea dañada.
- **ADAPTABILIDAD DEL ENTORNO:** Pretendemos que nuestra red de neuronas aprendan de la experiencia y que sea capaz de generalizar conceptos a partir de casos particulares.

Para ello, podemos establecer una estructura jerárquica que simule a la red neuronal biológica donde el elemento principal será la neurona artificial y se organizará en capas, donde varias capas constituirán una red neuronal.

2.3.2. Elementos de una Neurona Artificial

A continuación, en la Figura 2.2 describiremos los elementos que componen una neurona artificial según el modelo estándar, introducido por Rumelhart, McClelland,

2.3. ¿QUÉ ES UNA RED DE NEURONAS ARTIFICIALES?

y the PDP reserarch group (1986). Gráficamente, podemos ver la estructura de la i -ésima neurona y los elementos que la componen:

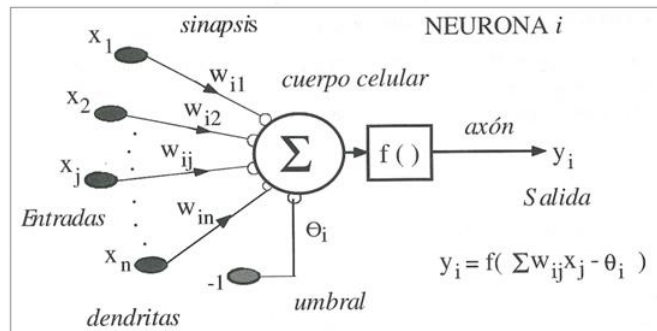


Figura 2.2: Estructura de una neurona artificial

1. Datos de entrada: $x_j, j = 1 \dots n$.
2. Pesos sinápticos: $w_{ij}, j = 1 \dots n$.
3. Regla de propagación: $h_i(x_1, \dots, x_n, w_{i1}, \dots, w_{in})$, definida a partir de los datos de entrada y de los pesos sinápticos. La regla de propagación más utilizada habitualmente consiste en combinar linealmente las entradas y los pesos sinápticos, obteniéndose:

$$h_i(x_1, \dots, x_n, w_{i1}, \dots, w_{in}) = \sum_{j=1}^n w_{ij}x_j.$$

Habitualmente, se suele añadir al conjunto de pesos sinápticos un parámetro θ_i , denominado *umbral*, de la siguiente manera:

$$h_i(x_1, \dots, x_n, w_{i1}, \dots, w_{in}) = \sum_{j=1}^n w_{ij}x_j - \theta_i.$$

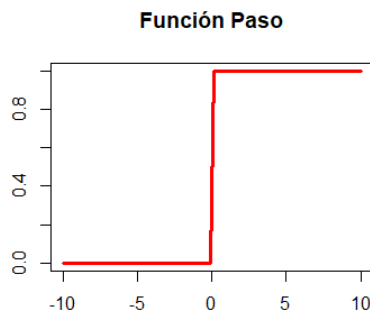
Denotando $x_0 = -1$ y $w_{i0} = \theta_i$, la expresión anterior se puede expresar de la siguiente forma:

$$h_i(x_1, \dots, x_n, w_{i1}, \dots, w_{in}) = \sum_{j=0}^n w_{ij}x_j = \sum_{j=1}^n w_{ij}x_j - \theta_i.$$

4. Función de activación: representa simultáneamente la salida de la neurona y su estado de activación. Denotando como y_i dicha función de activación, se tiene $y_i = f_i(h_i) = f_i(\sum_{j=0}^n w_{ij}x_j)$.

A continuación se verán varios ejemplos de funciones de activación. La función de activación *Paso* y *Signo* son ambas funciones escalonadas, en cambio las funciones *Logística* y *Tangente hiperbólica* son continuas.

■ **Función Paso:**

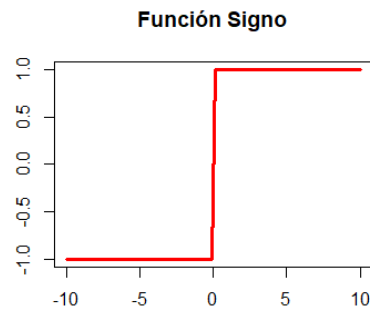


Se puede observar que la función de activación *Paso* es una función escalonada, y como consecuencia, la salida de la neurona será discreta, con valor 0 o 1.

$$y_i = \begin{cases} 1 & \text{si } \sum_{j=1}^n w_{ij}x_j \geq \theta_i; \\ 0 & \text{si } \sum_{j=1}^n w_{ij}x_j < \theta_i. \end{cases}$$

■ **Función Signo:**

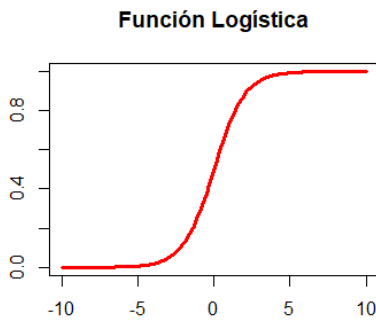
Al igual que en la función *Paso*, la función de activación *Signo* es una función escalonada con salida discreta.



$$y_i = \begin{cases} 1 & \text{si } \sum_{j=1}^n w_{ij}x_j \geq \theta_i; \\ -1 & \text{si } \sum_{j=1}^n w_{ij}x_j < \theta_i. \end{cases}$$

A continuación se verán funciones de activación continuas y que por consiguiente, tendrán una salida continua.

■ **Función Logística:**

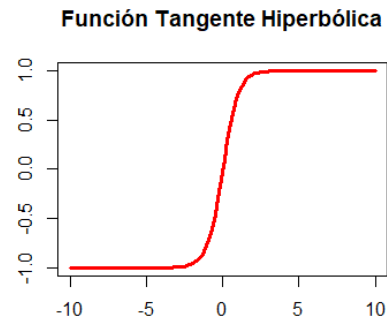


A diferencia de las dos funciones anteriores, se puede observar que la función de activación *Logística* es continua y por consiguiente tendrá una salida continua.

$$y_i = \frac{1}{1 + e^{-(\sum_{j=1}^n w_{ij}x_j - \theta_i)}}, \text{ con } y_i \in [0, 1].$$

■ **Función Tangente Hiperbólica:**

Como en el caso de la función *Logística*, se puede observar que la función de activación *Tangente Hiperbólica* es continua y por consiguiente tendrá una salida continua.



$$y_i = \frac{1}{1 + e^{-(\sum_{j=1}^n w_{ij}x_j - \theta_i)}}, \quad \text{con } y_i \in [0, 1].$$

Las dos últimas funciones de activación son continuas y diferenciables. El hecho de que sean diferenciables puede venir impuesto por la regla de aprendizaje, como será posteriormente, la *regla de retropropagación del error*. Hay que observar que las dos últimas funciones también son utilizadas en el perceptrón multicapa, que se explicará más adelante.

2.3.3. Redes neuronales

Una red neuronal se puede definir como un grafo dirigido (ver Figura 2.3), donde las neuronas (nodos) están estructuradas en distintas capas, entre las que se distinguen, la capa de entrada, la capa de salida y las capas ocultas. El grafo que representa la red neuronal tiene la siguientes propiedades:

1. A cada nodo (neurona) i se le asocia una variable de estado X_i .
2. A cada conexión entre la i -ésima neurona de una capa y la j -ésima neurona de la siguiente, denotado como conexión (i, j) se le asocia un peso $w_{ij} \in \mathbb{R}$.
3. A cada nodo (neurona) i se le asocia un umbral $\theta_i \in \mathbb{R}$.

4. Para cada nodo i se define una función de activación $f_i(x_1, \dots, x_n, w_{i1}, \dots, w_{in}, \theta_i)$ que depende tanto de los pesos sinápticos de sus conexiones, de los estados de los nodos j que están conectados con el nodo i y del umbral. El nuevo estado del nodo viene determinado por el valor de esta función.

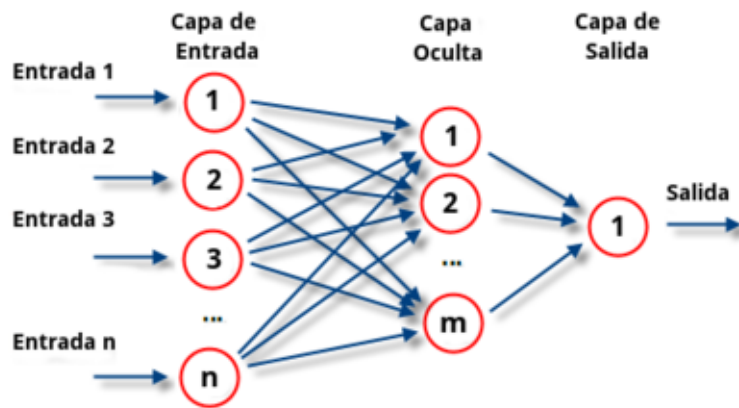


Figura 2.3: Red neuronal representada por un grafo dirigido

■ Tipos de redes neuronales

Una red neuronal puede ser *unidireccional* si no presenta bucles cerrados o conexiones, o *recurrente*, si el flujo de información tiene un bucle.

■ Reglas de aprendizaje

Se denomina *regla de aprendizaje* de una Red Neuronal Artificial a cualquier algoritmo que permita obtener una asignación de valores para cada uno de los coeficientes sinápticos.

Se puede distinguir entre *Aprendizaje Supervisado* y *Aprendizaje No Supervisado*.

- *Aprendizaje supervisado*: En este caso disponemos de un conjunto de datos de entrada y salida $D = \{(x^r, y^r), r = 1, \dots, N\}$. Al disponer de datos de

salida, se pueden mejorar los pesos sinápticos en la etapa $(m + 1)$ -ésima de la siguiente manera:

$$w_{ij}(m + 1) = w_{ij}(m) + \sum_{r=1}^N \Delta w_{ij}^r(m),$$

donde Δw_{ij}^r puede ser distinto según el caso, por ejemplo, en el caso del *Perceptrón Simple* introducido por Rosenblatt (1962) se tiene que:

$$\Delta w_{ij}^r(m) = \varepsilon(c_i^r - y_i^r)x_j^r, \quad (2.3.1)$$

donde $0 < \varepsilon < 1$ es el denominado *ritmo de aprendizaje* y c^r son las salidas respectivas en dicha etapa.

- *Aprendizaje No Supervisado*: En este caso no disponemos de datos de salida. Ahora, la red aprende a adaptarse basándose en las experiencias recogidas de los patrones de entrenamiento anteriores.

2.4. El Perceptrón Simple

A lo largo de este apartado se introducirá el *Perceptrón Simple* introducido por Roseblatt (1962). Se trata de un modelo unidireccional compuesto por una capa de entrada y otra de salida. Suponiendo que el perceptrón tiene n neuronas de entrada y m neuronas de salida, la i -ésima salida puede expresarse como:

$$y_i = f\left(\sum_{j=0}^n w_{ij}x_j\right) = f\left(\sum_{j=1}^n w_{ij}x_j - \theta_i\right), \text{ con } i = 1, \dots, m,$$

donde cabe observar que $x_0 = -1$ y $w_{i0} = \theta_i$, que las neuronas de entrada son discretas y la función de activación es escalonada.

La importancia del *Perceptrón Simple* radica en la capacidad del algoritmo de aprendizaje de determinar automáticamente los pesos sinápticos que clasifican un conjunto de patrones a partir de un conjunto de ejemplos etiquetados, o *conjunto de entrenamiento*. Es importante comentar que el perceptrón simple presenta grandes limitaciones, ya que tan sólo es capaz de discriminar entre dos clases linealmente separables.

El algoritmo de aprendizaje del *Perceptrón simple* se basa en la corrección de errores; es decir, ajusta los pesos de manera proporcional a la diferencia existente entre la salida de la red neuronal y la salida deseada, teniendo como objetivo minimizar el error actual de la red.

Para ver el funcionamiento del algoritmo de aprendizaje del Perceptrón Simple propuesto por Roseblatt (1962), utilizaremos un conjunto de entrenamiento, como el que se ha introducido en el apartado anterior de *Reglas de aprendizaje*. Sea $D = \{(x^r, y^r), r = 1, \dots, N\}$ dicho conjunto de entrenamiento. Se denota c^r a las salidas del perceptrón simple con $r = 1, \dots, N$. Supongamos además que tanto las variables de entrada como de salida toman sólo -1 y 1 como posibles valores. Como se explicó en las *reglas de aprendizaje*, los pesos se van actualizando de la siguiente manera:

- Si la salida c^r coincide con la salida deseada y^r no se actualizarán los pesos.
- En cambio, si la salida procedente del Perceptrón Simple no coincide con la deseada, los pesos nuevos serán:

$$w_{ij}^r = w_{ij}^r + \Delta w_{ij}^r, \text{ con } r = 1 \dots, N,$$

donde:

$$\Delta w_{ij}^r(m) = \begin{cases} 2\varepsilon c_i^r x_j^r & \text{si } y_i^r \neq c_i^r; \\ 0 & \text{si } y_i^r = c_i^r. \end{cases}$$

La actualización de los pesos, también se puede expresar de la siguiente manera:

$$\Delta w_{ij}^r(m) = \varepsilon(c_i^r - y_i^r)x_j^r. \quad (2.4.1)$$

Lo cual es más habitual para expresar la regla de actualización de pesos en este caso. Cabe destacar que en este caso, las actualizaciones de los pesos sólo podrán tomar los valores -2ε , 0 y 2ε , donde ε denota el *ritmo de aprendizaje*, un valor bajo implica un aprendizaje lento, mientras que un valor excesivamente grande puede conllevar oscilaciones excesivas de los pesos, no aconsejables en el proceso de entrenamiento.

Como se puede observar en la ecuación anterior, el proceso de aprendizaje es iterativo. Se comienza con una configuración sináptica inicial, normalmente pesos pequeños aleatorios presentándose los datos del conjunto de entrenamiento una y otra vez, de manera que los pesos se vayan ajustando de forma iterativa según la regla descrita anteriormente en 2.4.1, hasta que consigamos unos pesos que clasifique bien todo el conjunto de entrenamiento (si es posible).

El ajuste de los pesos en la iteración m debido a todo el conjunto de entrenamiento es:

$$w_{ij}(m+1) = w_{ij}(m) + \sum_{r=1}^N \Delta w_{ij}^r(m).$$

El algoritmo anterior siempre converge en un tiempo finito y con independencia de los pesos de partida, siempre que la función a representar sea linealmente separable. En cambio, si no es linealmente separable, el proceso de entrenamiento oscilará. Este hecho lo demostró Rosenblatt (1962).

2.5. El Perceptrón Multicapa

En este apartado se verá *El Perceptrón Multicapa* introducido por Rumelhart y cols. (1986). Un *Perceptrón Multicapa* está compuesto por una capa de entrada, una o varias capas ocultas y una capa de salida. Hay que observar que aunque se ha demostrado que una capa oculta cubre un espectro grande de problemas, se puede utilizar más de una (ver Figura 2.4).

Rumelhart y cols. (1986) formalizó el método *backpropagation error*, que permitía el aprendizaje de una red neuronal del tipo Perceptrón Multicapa facilitándole un conjunto de entradas con sus correspondientes salidas. El método tiene capacidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento.

A continuación, se explicará detalladamente el *Algoritmo backpropagation* aplicado a un *Perceptrón Multicapa* con 3 capas, la capa de entrada, la de salida y una capa oculta. De manera resumida, el algoritmo tiene dos etapas, una etapa

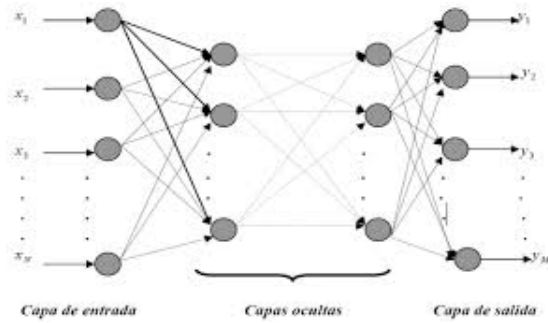


Figura 2.4: Perceptrón Multicapa con varias capas ocultas.

de funcionamiento, donde se presenta ante la red neuronal un patrón de entrada, y éste se transmite a través de las sucesivas capas de neuronas hasta obtener una salida, y una etapa de aprendizaje donde se modifican los pesos de la red de manera que coincida la salida obtenida por la red con la salida deseada.

Sea un *Perceptrón Multicapa* con 3 capas, donde las entradas, salidas, pesos y umbrales de las neuronas han sido definidos previamente. Dado un patrón de entrada x^r ($r = 1, \dots, N$), la operación global se representa del siguiente modo, para cada una de las k con ($k = 1, \dots, s$) neuronas de salida:

$$z_k^r = \sum_{j=1}^o w'_{kj} y_j^r - \theta'_k = \sum_{j=1}^o w'_{kj} f\left(\sum_{i=1}^n w_{ji} x_i^r - \theta_j\right) - \theta'_k.$$

Se parte del Error cuadrático medio como función de costo,

$$E(w, w', \theta, \theta') = \frac{1}{2} \sum_{r=1}^N \sum_{k=1}^m (c_k^r - z_k^r)^2.$$

Se minimizará con el método de descenso

por el gradiente, existiendo en este caso dos gradientes, uno respecto a la capa de salida, $\Delta w'_{kj} = -\varepsilon \frac{\partial E}{\partial w'_{kj}}$, y otro respecto a la capa oculta, $\Delta w_{ji} = -\varepsilon \frac{\partial E}{\partial w_{ji}}$.

A continuación, se desarrollarán las expresiones de actualización de los pesos, aplicando la regla de la cadena.

Primero, se puede observar que

$$\begin{aligned} E(w, w', \theta, \theta') &= \frac{1}{2} \sum_{r=1}^N \sum_{k=1}^m (c_k^r - z_k^r)^2 = \\ &= \frac{1}{2} \sum_{r=1}^N \sum_{k=1}^m (c_k^r - (\sum_{j=1}^o w'_{kj} f(\sum_{i=1}^n w_{ji} x_i^r - \theta_j) - \theta'_k))^2. \end{aligned}$$

Ahora, se desarrollan los dos gradientes:

$$\Delta w'_{kj} = -\varepsilon \frac{\partial E}{\partial w'_{kj}} = \varepsilon \sum_{r=1}^N (c_k^r - (\sum_{j=1}^o w'_{kj} y_j^r - \theta'_k)) y_j^r.$$

$$\begin{aligned} \Delta w_{ji} &= -\varepsilon \frac{\partial E}{\partial w_{ji}} = \\ &= \varepsilon \sum_{r=1}^N \sum_{k=1}^m (c_k^r - (\sum_{j=1}^o w'_{kj} f(\sum_{i=1}^n w_{ji} x_i - \theta_j) - \theta'_k)) w'_{kj} \frac{\partial f(\sum_{i=1}^n w_{ji} x_i - \theta_j)}{\partial (\sum_{i=1}^n w_{ji} x_i \theta_j)} x_i. \end{aligned}$$

La actualización de los umbrales (*bias*) se realiza por medio de las expresiones anteriores, hay que recordar que el umbral es un caso particular de un peso sináptico.

Se observa que el concepto clave de propagación hacia atrás de los errores está implícito. Primero se calcula la expresión denominada *señal de error*, con la que se calcula la actualización $\Delta w'_{kj}$.

$$\text{señal de error} \equiv (c_k^r - (\sum_{j=1}^o w'_{kj} y_j^r - \theta'_k)).$$

Después se propagan hacia atrás los errores a través de la sinapsis de la capa oculta y con estas señales de error se calcula la actualización de Δw_{ji} .

$$\Delta w_{ji} = \sum_{k=1}^s (c_k^r - (\sum_{j=1}^o w'_{kj} f(\sum_{i=1}^n w_{ji} x_i - \theta_j) - \theta'_k)) w'_{kj} \frac{\partial f(\sum_{i=1}^n w_{ji} x_i - \theta_j)}{\partial (\sum_{i=1}^n w_{ji} x_i \theta_j)}.$$

Una vez descrito las componentes necesarias, el procedimiento para entrenar nuestro *Perceptrón Multicapa* mediante el algoritmo *Backpropagation* es el siguiente:

Algorithm 1 Algoritmo de Backpropagation

- 1: Se establecen aleatoriamente pesos y umbrales iniciales ($t = 0$).
 - 2: Para cada patrón r del conjunto de entrenamiento:
 - 3: Se lleva a cabo una fase de ejecución para obtener la respuesta frente al patrón r -ésimo utilizado.
 - 4: Se calculan las señales de error asociadas.
 - 5: Se calcula el incremento parcial de los pesos y umbrales debidos a cada patrón r .
 - 6: Se calcula el incremento total de todos los patrones sobre los pesos $\Delta w'_{kj}$ y Δw_{ji} . Al igual que con los umbrales.
 - 7: Se actualizan dichos pesos y umbrales.
 - 8: Se calcula el error total.
 - 9: Se actualiza $t = t + 1$ y se vuelve al paso 2 si todavía el error total no es satisfactorio.
-

Hay que destacar que se debe comenzar siempre con pesos iniciales aleatorios pequeños, ya sean positivos o negativos. En el esquema anterior se lleva a cabo una fase de ejecución para todos y cada uno de los patrones del conjunto de entrenamiento, se calcula la variación de los pesos debido a cada patrón, se acumulan y a continuación se efectúa la actualización de los pesos.

2.6. *Deep Learning*

2.6.1. Introducción

Dentro del estudio de las redes neuronales se encuentra el campo del *Deep Learning* (*DL*), que utiliza arquitecturas de red más profundas (con un mayor número de capas y de nodos) para clasificación y regresión. Básicamente el comportamiento es similar a lo anteriormente explicado, con la única diferencia que implementa nuevos algoritmos que mejoran su rendimiento. Pero como ya se ha comentado, lo que realmente proporciona un mayor rendimiento es el mayor número de capas de su estructura. A continuación veremos las características principales que justifican su éxito, junto a una justificación matemática de algunas de ellas, como son la optimalidad global, la estabilidad geométrica, etc, (véase Vidal, Bruna, Giryes, y Soatto (2017)).

En particular, hay tres factores fundamentales en el *DL*, la arquitectura, las técnicas de “regularización” y los algoritmos de optimización, los cuales son fundamentales para entrenar una red neuronal profunda que tenga un buen rendimiento. Por tanto, entender la necesidad de estos factores es esencial para comprender su éxito:

- **Aproximación:** Una propiedad esencial en el diseño de la arquitectura de una red neuronal es su habilidad para aproximar funciones arbitrarias sobre los datos de entrada. Aunque ya vimos anteriormente que una red neuronal con una capa oculta es un aproximador de funciones universal, se observa que arquitecturas profundas son capaces de mejorar la captación de propiedades invariantes de los datos en comparación con la redes neuronales

convencionales.

- **Generalización y regularización:** Otra propiedad esencial de las arquitecturas de las redes neuronales es su habilidad para generalizar con un número pequeño de muestras de entrenamiento. Las redes neuronales profundas son entrenadas con muchos menos datos que número de parámetros y aun así pueden prevenir el sobreajuste utilizando técnicas de regularización muy simples, como es por ejemplo el *Dropout*, que básicamente consiste en coger un subconjunto aleatorio de parámetros en cada iteración.
- **Facilidad para la detección de los patrones importantes:** Una de las propiedades más importantes que tienen estos métodos son la facilidad para detectar la información importante y también los criterios que son invariantes entre los casos similares, lo cual es esencial para predecir observaciones futuras.
- **Optimización:** Versiones modernas del algoritmo de *Backpropagation* residen en el *Stochastic Gradient Descent* (SGD) para aproximar el gradiente a conjuntos de datos enormes. Hay que observar, que mientras el SGD ha sido muy analizado para funciones de pérdida convexas, en *DL* la función de pérdida es no convexa respecto a los pesos de la red neuronal, por lo que no hay garantías de que SGD encuentra el mínimo global. Más adelante se verán trabajos recientes que muestran que el error para problemas de optimización multidimensionales y no convexas como los que presenta *DL* tiene propiedades beneficiosas.

2.6.2. Optimalidad global en *DL*

El funcionamiento básico de una red neuronal *DL* es el mismo que el de una red neuronal convencional, el cambio reside en la función de pérdida, o la función a minimizar para encontrar los pesos óptimos. El problema de la obtención de pesos se formula como el siguiente problema de optimización:

$$\min_{\{W^k\}_{k=1}^K} l(Y, \phi(X, W^1, \dots, W^K)) + \lambda \Theta(W^1, \dots, W^K), \quad (2.6.1)$$

donde $l(Y, \phi)$ es la función pérdida que mide el ajuste entre la verdadera salida, Y , y la salida que ha predicho el modelo, $\phi(X, W)$. Hay que observar que W^k denotan los pesos de la capa k -ésima, con $k \in \{1, \dots, K\}$. Θ es la función de regularización que previene el sobreajuste, por ejemplo, si tomamos l_2 , $\Theta(W) = \sum_{k=1}^K \|W^k\|_F^2$, con $\lambda \geq 0$ el parámetro de equilibrio.

2.6.3. El reto de las funciones no convexas en las redes neuronales

El principal reto en el problema de optimización de los pesos en 2.6.1 es la *no convexidad*, incluso siendo la función de penalización o pérdida $l(Y, \phi)$ normalmente convexa respecto a ϕ , la función $\phi(X, W)$ no suele ser una función convexa respecto a los pesos W debido al producto entre las variables W^k y las funciones de activación. Sin embargo, en los problemas no convexos, el conjunto de puntos críticos incluyen no sólo mínimos globales, también mínimos locales, máximos locales, puntos de silla. Por tanto, la no convexidad del problema deja la cuestión un poco mal planteada en el sentido de que no sólo la formulación del modelo es

importante, también los detalles de implementación, como pueden ser, con qué pesos el modelo se inicia y sobre todo algoritmos de optimización de los mismos. Dichas implementaciones “al margen” del ejercicio es crucial en el rendimiento del modelo. Para solventar el problema de la no convexidad, una estrategia ampliamente utilizada en los modelos *DL* es inicializar los pesos de manera aleatoria, actualizar dichos pesos con el método del gradiente local, comprobar que el error decrece lo suficientemente rápido, y si no, tomar otros pesos para la inicialización.

2.6.4. Estabilidad geométrica en el *DL*

Una cuestión importante a tener en cuenta son las suposiciones matemáticas que utilizan los métodos de *DL* y que hacen que en determinados casos funcione correctamente, o al menos mejor que los métodos clásicos. Sea $\Omega = [0, 1]^d$ un dominio Euclídeo compacto d -dimensional donde están definidas funciones $X \in L(\Omega)^2$ (véase Def. 2.6.1). En aprendizaje supervisado, una función no conocida $f : L(\Omega)^2 \rightarrow Y$ es observada en el conjunto de entrenamiento

$$\{X_i \in L(\Omega)^2, Y_i = f(X_i)\}_{i \in I}, \quad (2.6.2)$$

donde el espacio de llegada Y puede ser considerado discreto con $C = |Y|$ el número de clases de las que dispone la variable de salida, o $Y = \mathbb{R}^C$ en un problema de regresión.

Definición 2.6.1. *Sea X un espacio de medida y μ una medida positiva. Se define el espacio de funciones de potencia p -ésima integrable como:*

$$L^p(\mu) = \left\{ f : X \rightarrow \mathbb{C} : f \text{ medible y } \int_X |f|^p d\mu < \infty \right\}.$$

En la mayoría de los casos, se considera que la función f satisface las siguientes condiciones:

1. **Estacionariedad:** Se considera un operador de traslación

$$\tau_v X(u) = X(u - v), \quad u, v \in \Omega, \quad (2.6.3)$$

que actúa sobre funciones $X \in L(\Omega)^2$. Dependiendo del problema, se asume que la función f es *invariante* respecto a las traslaciones. Formalmente, se tiene que $f(\tau_v X) = f(X)$ para cualquier $X \in L(\Omega)^2$ y $v \in \Omega$.

2. **Deformaciones locales y separaciones a escala:** De forma similar, una deformación \mathcal{L}_τ , donde $\tau : \Omega \rightarrow \Omega$ es un vector de funciones de clase C^∞ que actúa sobre $L(\Omega)^2$ como $\mathcal{L}_\tau X(u) = X(u - \tau(u))$. Las deformaciones pueden modelar traslaciones locales, rotaciones. La mayoría de los problemas que se estudian no son solamente invariantes respecto a traslaciones, además son estables respecto a deformaciones locales. En problemas que son invariantes respecto a la traslación se tiene:

$$|f(\mathcal{L}_\tau X) - f(X)| \approx \|\nabla \tau\|, \quad (2.6.4)$$

para todo X, τ , donde $\|\nabla \tau\|$ mide la derivabilidad de una deformación dada; es decir, cuantifica si las predicciones no cambian mucho si la entrada es deformada localmente. Esta propiedad es mucho más fuerte que la estacionariedad.

Una vez explicados los aspectos fundamentales del *DL* y los aspectos básicos de su buen rendimiento en general, se va a intentar aplicar dicho método a un caso práctico del baloncesto, hay que tener en cuenta, que se han tomado pesos

sinápticos iniciales bajos pero que no se ha optimizado ni el número de capas ni el conjunto de pesos iniciales, aún así, el resultado roza el 90 % de tasa global de acierto, lo que no se ha conseguido con ninguna de las redes neuronales que no utilizan el método del *DL*.

2.7. Aplicaciones al baloncesto

2.7.1. Objetivos

El objetivo principal de esta sección es aplicar los resultados anteriores al Baloncesto. Utilizaremos *RStudio* para realizar los cálculos, junto con varias librerías especializadas en Redes Neuronales, como son por ejemplo, *caret*, *nnet*, *h2o*. Los datos de los que disponemos son oficiales de la empresa Sport-VU en 2012, obtenidos de la página web Rajiv Shah (22 de Agosto de 2013), ya que hasta ese mismo año, la NBA ofrecía gratuitamente los datos de *tracking* de los partidos, por lo que se ha aprovechado esos datos para realizar el estudio. Dichos datos se describirán en el siguiente apartado. Aunque hay muchísimas aplicaciones de las Redes Neuronales al baloncesto, empezaremos con un ejemplo sencillo, ya que disponemos de potencia computacional limitada y se tarda mucho en obtener los resultados. Por lo tanto, en principio haremos una predicción de las distintas posiciones de los jugadores de baloncesto según unas variables que explicaremos en la siguiente sección. Por último, daremos posibles líneas por las que puede continuar aplicándose estos métodos.

2.7.2. Datos

Los datos de los que disponemos son del partido entre los Detroit Pistons y los Atlanta Hawks el 27 de Octubre de 2017 con un resultado final de 106-94 a favor de los Detroit Pistons (véase Fig. 2.5). Cada registro tiene 13 columnas:

1. *player_id*: El código que identifica a cada jugador.
2. *lastname*: Apellido del jugador.
3. *firstname*: Nombre del jugador.
4. *jersey*: Número del dorsal.
5. *position*: Posición en la que juega dicho jugador, hay cinco, **C** como Pívot, **F-C** como Ala-Pívot, **F** como Alero, **G** como Base y **G-F** como Escolta.
6. *team_id*: El código que identifica a cada equipo.
7. *x_loc*: Coordenada X de la posición de dicho jugador en ese instante.
8. *y_loc*: Coordenada Y de la posición de dicho jugador en ese instante.
9. *radius*: Distancia al aro cuando se realiza el tiro.
10. *game_clock*: El tiempo en segundos en el que se ha producido la acción desde el inicio del cuarto. Va desde cero a 720 segundos, que son los 12 minutos que dura dicho cuarto.
11. *shot_clock*: El tiempo en ese instante que llevan de posesión en segundos (entre 0 y 24 segundos).
12. *quarter*: El cuarto en el que se da la acción. Va desde 1 a 4.
13. *event.id*: El tipo de evento que se está produciendo en ese momento. No se dispone de un anexo que indique qué es cada evento.

	player_id	lastname	firstname	jersey	position	team_id	x_loc	y_loc	radius	game_clock	shot_clock	quarter	event.id
1	201952	Teague	Jeff	0	G	1610612737	28.26728	23.73400	0	720	24	1	1
2	201952	Teague	Jeff	0	G	1610612737	28.35063	23.71423	0	720	24	1	1
3	201952	Teague	Jeff	0	G	1610612737	28.43244	23.69250	0	720	24	1	1
4	201952	Teague	Jeff	0	G	1610612737	28.50975	23.66888	0	720	24	1	1
5	201952	Teague	Jeff	0	G	1610612737	28.58962	23.64730	0	720	24	1	1
6	201952	Teague	Jeff	0	G	1610612737	28.66770	23.62981	0	720	24	1	1

Figura 2.5: Datos de SportsVU del partido Detroit Pistons - Atlanta Hawks

Hay que observar que ahora se tienen que tratar para nuestro objetivo, que se rá predecir en qué posición juega cada jugador utilizando la posición en la que se encuentra, tanto en el eje OX como en el eje OY (variables x_loc e y_loc), también utilizaremos el tiempo de partido junto con el tiempo de posesión ($game_clock$ y $shot_clock$ respectivamente), y por último se utilizará también la variable $event.id$, que es un identificador de la jugada que se está haciendo en ese momento junto con el cuarto ($quarter$) en el que se está desarrollando el juego. Es obvio que no se debe utilizar el nombre del jugador ni su identificador, ya que determinarían directamente la posición en la que juega dicho jugador (véase Fig. 2.6).

	position	x_loc	y_loc	game_clock	shot_clock	quarter	event.id
1	G	0.3375914	0.4798811	1	1	0	0
2	G	0.3383630	0.4795169	1	1	0	0
3	G	0.3391204	0.4791166	1	1	0	0
4	G	0.3398361	0.4786815	1	1	0	0
5	G	0.3405755	0.4782839	1	1	0	0
6	G	0.3412984	0.4779617	1	1	0	0

Figura 2.6: Datos para realizar el estudio

2.7.3. Tratamiento de los datos

Para que no haya problemas en el estudio, se necesita que los datos sean consistentes y que estén tratados de la manera correcta para realizar un buen modelo. Para ello, se pasa la columna a predecir a factor y se normalizan las demás columnas (véase Fig. 2.7).

```
datos<-na.omit(datos)
datos$position<-factor(datos$position)

maximos= apply(datos[,2:7],2,max)
minimos= apply(datos[,2:7],2,min)
datos[,2]= (datos[,2]-minimos[1])/(maximos[1]-minimos[1])
datos[,3]= (datos[,3]-minimos[2])/(maximos[2]-minimos[2])
datos[,4]= (datos[,4]-minimos[3])/(maximos[3]-minimos[3])
datos[,5]= (datos[,5]-minimos[4])/(maximos[4]-minimos[4])
datos[,6]= (datos[,6]-minimos[5])/(maximos[5]-minimos[5])
datos[,7]= (datos[,7]-minimos[6])/(maximos[6]-minimos[6])

datos[,1]<-as.factor(datos[,1])
```

Figura 2.7: Tratamiento de los datos

Una vez transformado el conjunto de datos, se procede a dividir el conjunto de datos en *entrenamiento* y *test*, el primero se utilizará para obtener el modelo y el segundo para testarlo. Normalmente, se divide el conjunto de datos en 2 partes, el 75% de los datos para el conjunto de entrenamiento y los restantes para el conjunto test (véase Fig. 2.8).

```
n<-nrow(datos)
indices<-1:n
indient<-sample(indices,0.75*n)
inditest<-setdiff(indices,indient)

entrenamiento<-datos[indient,]
test<-datos[inditest,]
```

Figura 2.8: División en conjunto entrenamiento y conjunto test

2.7.4. Aplicaciones

En esta sección, se aplicarán redes neuronales con métodos de *DL* a los datos explicados anteriormente, de cara a predecir la posición de los jugadores según las variables que hemos descrito anteriormente. Para ello, utilizaremos la librería *h2o* que utiliza las redes neuronales y el aprendizaje profundo, donde los detalles se explicarán más adelante en esta sección. Después de varios modelos, hemos

obtenido un modelo que obtiene una *tasa de acierto global* del 87%. Dicho modelo se explicará a continuación (véase Fig. 2.9).

```
modelo_h2o5 <- h2o.deeplearning(  
  x = 2:7, y = 1,  
  training_frame = train.hex,  
  validation_frame=test.hex,  
  distribution="multinomial",  
  activation = 'RectifierWithDropout',  
  hidden = c(700,500,500,400),  
  hidden_dropout_ratio = rep(0.1,4),  
  input_dropout_ratio = 0.2,  
  epochs = 75,  
  l1 = 1e-5,  
  l2 = 1e-5,  
  rho = 0.99,  
  epsilon = 1e-8,  
  train_samples_per_iteration = 8000)  
summary(modelo_h2o5)
```

Figura 2.9: Modelo de clasificación

En la Fig. 2.9, se ha utilizado el comando `h2o.deeplearning` del paquete `h2o` que construye una red neuronal profunda utilizando redes neuronales artificiales con el algoritmo de *back propagation* y con la siguientes propiedades:

1. *x*: El número de columnas que utilizamos como variables de entrada, en nuestro caso se corresponden con las columnas 2 a 7.
2. *y*: La variable de salida, en nuestro caso es la columna 1, que corresponde a la posición en la que juega cada jugador.
3. *training_frame*: El conjunto de entrenamiento que se va a utilizar para “entrenar” a nuestra red neuronal, en nuestro caso corresponde con el conjunto de datos *train.hex*.
4. *validation_frame*: El conjunto test que se va a utilizar para realizar las predicciones y por tanto, para obtener la eficiencia del modelo que se ha creado.

En este caso, se utilizará el conjunto de datos *test.hex*.

5. *distribution*: Aquí se describe la función de distribución, en nuestro caso utilizaremos la función *Multinomial* ya que la salida es de tipo factor con 5 niveles distintos.
6. *activation*: Función de activación que se utiliza en el modelo, en nuestro caso se utiliza *RectifierWithDopout*.
7. *hidden*: Se introduce un vector con el número de nodos que posee cada capa, en nuestro caso se tienen 4 capas con 700, 500, 500 y 400 nodos respectivamente.
8. *hidden_dropout_ratio*: Los pesos iniciales para las capas ocultas, en nuestro caso se utilizará $w = 0,1$ para cada una de las capas ocultas.
9. *input_dropout_ratio*: El peso inicial de la capa de entrada y el de la capa de salida, en este caso se utilizará $w = 0,2$.
10. *epochs*: En este apartado se indica cuántas veces debemos iterar nuestro conjunto de datos, en nuestro caso, se utilizará 75. Hay que observar que un número grande puede sobreajustar los datos y no nos conviene.
11. *l1*: Regularización L1, lo que añade estabilidad e incrementa la efectividad (ya que hace que muchos pesos vayan a cero). En este caso, se utilizará 10^{-5} .
12. *l2*: Regularización L2, lo que añade estabilidad e incrementa la efectividad (ya que hay muchos pesos que son demasiado pequeños). En este caso, se utilizará 10^{-5} también.
13. *rho*: Ritmo de aprendizaje. Por defecto se utiliza 0,99.
14. *epsilon*: Ritmo de aprendizaje suavizado. Para evitar divisiones por cero y permitir que el proceso siga funcionando. Por defecto se utiliza 10^{-8}

15. *train_samples_per_iteration*: Número de muestras globales que se van a utilizar para reducir los tiempos de procesado, en este caso se utilizarán 8000 muestras de los datos que disponemos.

Con este modelo, hemos predicho el tipo de jugador del conjunto test, obteniendo la siguiente tabla de confusión (véase Fig. 2.10), donde en la fila aparece el tipo de jugador que es, y en la columna el tipo de jugador que el modelo ha predicho que es. Con este modelo obtenemos una tasa de acierto global del 87%.

	C	F	F-C	G	G-F	Acierto_test
C	33307	4030	2380	3563	413	76.2
F	1929	133758	4095	14686	1935	85.5
F-C	1460	3214	58306	5265	180	85.2
G	1716	8663	5238	208721	1503	92.4
G-F	526	2073	1045	2822	36098	84.8

Figura 2.10: Tabla de confusión y tasa de acierto por posición

2.7.5. Conclusiones

Después de muchos modelos, se ha obtenido un modelo que tiene una tasa de acierto bastante buena. Cabe destacar que en este modelo no se han utilizado técnicas para optimizar el número de capas y el número de nodos de la misma, ni tampoco hemos utilizado técnicas para optimizar los pesos iniciales, pero esto ya es bastante más avanzado y no se dispone de tiempo para presentarlo. Aún así la tasa de acierto es muy grande, aunque mejorable. Hay que observar que para realizar los cálculos computacionales necesarios se ha utilizado el servidor del Departamento de Estadística e Investigación Operativa de la Facultad de Matemáticas de la Universidad de Sevilla con la edición de Windows *Windows Server 2008 R2 Enterprise*, con las siguientes características en su sistema:

1. Procesador: Intel(R) Xeon(R) CPU E5-2620 v3 2.40GHz (2 procesadores)
2. Memoria RAM: 64,0 GB (63,9 GB utilizable)
3. Tipo de sistema: Sistema operativo de 64 bits

Gracias a este permiso he podido obtener una velocidad de cálculo mucho mayor que con un ordenador convencional, lo que me ha permitido obtener modelos con mejor rendimiento.

Posibles líneas futuras con redes neuronales son la clasificación no supervisada de las distintas jugadas de un partido (véase Kempe, Grunz, y Memmert (2015)). En el conjunto de datos que se dispone hay una columna que distingue los distintos tipos de eventos, pero no se dispone de un apéndice que indique qué tipo de jugada o evento es cada número, por lo que es complicado clasificarlo. La utilidad de dicha clasificación es enorme, dado que clasificaría rápidamente todas las jugadas sin la necesidad del factor humano y sobre todo en mucho menos tiempo, lo cual es muy importante en estos tipos de competiciones, ya que el tiempo es uno de los factores limitantes.

3 ANÁLISIS ESPACIAL

“Si nos sale bien habremos cambiado el juego para siempre”

Billy Beane (1962-)

3.1. Introducción

De manera general, los jugadores NBA son comparados entre sí mediante intuiciones poco objetivas y difíciles de medir sobre su comportamiento en la cancha. Ésto dificulta las comparaciones objetivas entre los jugadores y hacer predicciones realistas sobre dicho comportamiento. En este capítulo se pretende modelar los datos de intentos de tiro como un *Proceso Puntual o de puntos* y se creará una representación con un número bajo de dimensiones de los tipos de jugadores ofensivos en la NBA. Se utilizará una matriz de factorización no negativa (denotada como NMF), que es una técnica de reducción de la dimensión no supervisada, y se observará que esta descomposición resume los hábitos de tiro de los jugadores NBA. Las representaciones implementadas por el algoritmo describen distintos tipos de jugadores y puede ser utilizada para modelar otras características espaciales, como la precisión en el tiro (véase Miller, Bornn, Adams, y Goldsberry (2014)).

3.2. Resultados teóricos

En este apartado se explicarán resumidamente los distintos conceptos que tendremos en cuenta a lo largo del capítulo para desarrollar el estudio.

3.2.1. Procesos estocásticos

Definición 3.2.1. *Un proceso estocástico es una familia de variables aleatorias que, en general, están relacionadas entre sí y siguen una ley de distribución conjunta. Se denota por $\{Y_t\}_{t \in T}$.*

3.2.2. Procesos puntuales espaciales

Definición 3.2.2. *Un proceso puntual es un modelo estocástico que genera un número finito de eventos $\{x_i, i = 1, \dots, n\}$ en un conjunto X (con $X \subset \mathbb{R}^d$ acotado). Si los eventos tienen asociadas medidas o marcas, se tiene un proceso puntual con marcas. Si un evento situado en $x \in \mathbb{R}^d$ está asociado a una marca $s \in \mathcal{F}$, entonces (x, s) es un punto en $\mathbb{R}^d \times \mathcal{F}$. Del mismo modo, si observamos un proceso puntual con marcas en los instantes $(t_1, t_2, \dots, t_n) \in T$, se puede tomar como un proceso puntual en $\mathbb{R}^d \times \mathcal{F} \times T$, dando lugar a los procesos puntuales espacio-temporales.*

3.2.3. Procesos Gaussianos

Definición 3.2.3. *Un Proceso Gaussiano es un proceso estocástico $\{X\}_{t \in T}$ donde fijado t , se tiene que, $X_t \sim N(\mu_t, \sigma_t)$.*

Los Procesos Gaussianos son utilizados frecuentemente como modelos probabilísticos sobre funciones $f : \chi \rightarrow \mathbb{R}$, donde el valor $f_n \equiv f(x_n)$ corresponde a la función evaluada en algún punto $x_n \in \chi$. La covarianza espacial entre dos puntos en χ esconden pistas sobre un amplio aspecto de propiedades de f , como por ejemplo, diferenciabilidad, suavidad y periodicidad.

Como ejemplo, sea una función de clase C^∞ ; es decir, infinitamente derivable, $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ para la cual se ha observado un conjunto de localizaciones x_1, \dots, x_N y sus correspondientes valores f_1, \dots, f_N . Podemos modelizar la propiedad de la “suavidad” tomando una función de covarianza que resulta de procesos de suavización.

De hecho, la función de covarianza exponencial cuadrada

$$\text{cov}(f_i, f_j) = k(x_i, x_j) = \sigma^2 e^{-\frac{1}{2} \frac{\|x_i - x_j\|^2}{\phi^2}}, \quad (3.2.1)$$

asume que la función f es infinitamente diferenciable, con una variación marginal σ^2 y una longitud de escala ϕ que controla el número esperado de cambios de dirección de la función. Obsérvese que debido a que la covarianza es estrictamente una función de la distancia entre dos puntos del espacio χ , la función de covarianza exponencial cuadrada es estacionaria. Hay que observar que utilizaremos esta propiedad de la suavización para hacer que el sesgo de los hábitos de tiro varíe suavemente sobre la pista.

3.2.4. Procesos de Poisson

Se verán algunas definiciones necesarias para introducir conceptos de *Procesos de Poisson*, entre ellas se verá el concepto de *distribución de Poisson*, *Proceso de Poisson* y *Procesos de Poisson No Homogéneos*.

Definición 3.2.4. Una variable aleatoria X sigue una distribución de Poisson de parámetro $\lambda > 0$ si toma valores en el conjunto $\{0, 1, 2, \dots\}$, con probabilidad dada por

$$P(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}, \text{ para } k = 0, 1, 2, \dots$$

Definición 3.2.5. Sean T_1, T_2, \dots variables aleatorias independientes idénticamente distribuidas (v.a.i.i.d) con distribución exponencial de parámetro λ . Sea $\tau_0 = 0$ y $\tau_n = T_1 + \dots + T_n$ para $n \geq 1$. Definimos el Proceso de Poisson de parámetro (intensidad) λ por

$$N(s) = \max\{n : \tau_n \leq s\}, s \geq 0.$$

Las variables T_n representan los intervalos de tiempo entre eventos sucesivos

(llegada de clientes a una central telefónica, de pacientes a la emergencia de un hospital, etc.). $\tau_n = T_1 + \dots + T_n$ es el instante en el que ocurre el n -ésimo evento y $N(s)$ es el número de eventos que han ocurrido hasta el instante s .

Si el parámetro de un *Proceso de Poisson*, que representa la intensidad por unidad de tiempo con la que ocurren los eventos, no es constante a lo largo del tiempo se tendría el *Proceso de Poisson No Homogéneo*.

Definición 3.2.6. *Se define Proceso de Poisson No Homogéneo como un Proceso de Poisson cuyo parámetro, que representa la intensidad por unidad de tiempo con la que ocurren los eventos, no es constante a lo largo del tiempo. Sea $\{(N(t), t \geq 0)\}$ es un proceso de Poisson No Homogéneo con tasa $\lambda(s), s \geq 0$ si*

1. $N(0) = 0$,
2. $N(t)$ tiene incrementos independientes,
3. $N(s+t) - N(s)$ tiene distribución de Poisson con media $\int_s^{s+t} \lambda(r) dr$.

Proposición 3.2.7. *En este caso los intervalos de tiempo entre eventos sucesivos, $T_n, n \geq 1$, ya no son independientes ni tienen distribución exponencial.*

Demostración. Esta es la razón principal por la cual no utilizamos nuestra definición inicial de proceso de Poisson. Veamos ahora que la proposición es cierta. Sea $\mu(t) = \int_0^t \lambda(s) ds$, entonces $N(t) \sim Pois(\mu(t))$ y

$$P(T_1 > t) = P(N(t) = 0) = e^{-\mu(t)}.$$

Derivando, se obtiene la densidad

$$f_{T_1}(t) = -\frac{d}{dt}P(T_1 > t) = \lambda e^{-\int_0^t \lambda(s) ds} = \lambda(t)e^{-\mu(t)},$$

para $t \geq 0$. De manera similar podemos obtener la densidad conjunta

$$f_{T_1, T_2}(u, v) = \lambda(u)e^{-\mu(u)}\lambda(v)e^{-(\mu(v)-\mu(u))} = \lambda(u)\lambda(v)e^{-\mu(v)},$$

y haciendo ahora el cambio de variables $u = s$ y $v = s + t$

$$f_{T_1, T_2}(s, t) = \lambda(s)e^{-\mu(s)}\lambda(s+t)e^{-(\mu(s+t)-\mu(s))} = \lambda(s)\lambda(s+t)e^{-\mu(s+t)},$$

de modo que T_1 y T_2 no son independientes si $\lambda(s)$ no es constante. □

Ahora se irán introduciendo estos conceptos pero aplicados al baloncesto. Se definirá el proceso de Poisson como un proceso de puntos completamente aleatorios en un espacio χ , para el cual el número de puntos que se encuentran en un conjunto $A \subseteq \chi$ se distribuyen según una Poisson. Se utilizará un proceso de Poisson No Homogéneo en un dominio χ ; es decir, modelaremos el conjunto de puntos en el espacio, x_1, \dots, x_N con $x_n \in \chi$ como un proceso de Poisson con una función de intensidad no negativa $\lambda(x) : \chi \rightarrow \mathbb{R}_+$. Esto implica que para cualquier conjunto $A \subseteq \chi$, el número de puntos que caen en A , N_A , se distribuyen según una Poisson,

$$N_A \sim \text{Poiss}\left(\int_A \lambda dA\right).$$

Además, los procesos de Poisson no tienen memoria; es decir, N_A es independiente de N_B si A y B son conjuntos disjuntos.

Se denotará $\mathbf{x} \sim PP(\lambda(\cdot))$ que un conjunto de puntos $\mathbf{x} \equiv \{x_1, \dots, x_N\}$ sigue un proceso de Poisson.

Una propiedad muy útil de los procesos de Poisson es el teorema de superposición.

Teorema 3.2.8. *Sean $\mathbf{x}_1, \mathbf{x}_2, \dots$ un conjunto finito de procesos de Poisson independientes, con parámetros de intensidad $\lambda_1, \lambda_2, \dots$, su superposición está distribuida como*

$$\bigcup_{k=1}^{\infty} \mathbf{x}_k \sim PP\left(\sum_{k=1}^{\infty} \lambda_k\right).$$

La superposición de los procesos de Poisson motivan la descomposición no negativa de un proceso de Poisson global en subprocesos más pequeños que pueden ser compartido entre los jugadores.

3.2.5. Procesos Log-Gaussianos Cox

Definición 3.2.9. *Se define un proceso Log-Gaussiano Cox (LGCP) como un proceso estocástico de Poisson de segundo orden donde la función de intensidad varía según el espacio y está modelada según una exponencial de un proceso Gaussiano; es decir,*

$$\begin{aligned}Z(\cdot) &\sim GP(0, k(\cdot, \cdot)), \\ \lambda(\cdot) &\sim \exp(Z(\cdot)), \\ x_1, \dots, x_N &\sim PP(\lambda(\cdot)),\end{aligned}$$

donde de segundo orden se refiere a los dos niveles de aleatoriedad: en la función $Z(\cdot)$ que es un proceso Gaussiano visto en 3.2.3 y en el procesos de Poisson con intensidad $\lambda(\cdot)$.

3.2.6. Matriz de Factorización No Negativa

La factorización con una matriz no negativa es una técnica para reducir la dimensión que asume que una matriz Λ puede ser aproximada por el producto de dos matrices de bajo rango:

$$\Lambda = WB$$

donde la matriz $\Lambda \in \mathbb{R}_+^{N \times V}$ está compuesta por N puntos de longitud V , la *matriz base* $B \in \mathbb{R}_+^{K \times V}$ está compuesta por K vectores base, y la matriz de pesos $W \in \mathbb{R}_+^{N \times K}$ está compuesta por N vectores de pesos no negativos que escalan y combinan linealmente los vectores base para construir Λ .

Cada vector puede ser construido con los pesos y las bases de la siguiente manera

$$\lambda_n = \sum_{k=1}^K W_{n,k} B_{k,v}.$$

Las matrices óptimas W^* y B^* están determinadas por un proceso de optimización que minimiza $l(\cdot, \cdot)$, la medida del error entre WB y Λ con la restricción de que todos los elementos tienen que ser no negativos:

$$(W^*, B^*) = \arg \min_{W, B \geq 0} l(\Lambda, WB). \quad (3.2.2)$$

Diferentes medidas del error dan resultados diferentes. Para matrices arbitrarias X e Y , una opción es la norma de Frobenius,

$$l_2(X, Y) = \sum_{i,j} (X_{ij} - Y_{ij})^2.$$

Otra posibilidad es la medida para una matriz de divergencia

$$l_{KL}(X, Y) = \sum_{i,j} X_{ij} \log \frac{X_{ij}}{Y_{ij}} - X_{ij} + Y_{ij},$$

que se reduce a la divergencia de KullBack-Leibler (KL) cuando interpretamos las matrices X e Y como distribuciones discretas, y por consiguiente, $\sum_{ij} X_{ij} = \sum_{ij} Y_{ij} = 1$, (véase D. D. Lee y Seung (2001)). Hay que observar que minimizar la divergencia $l_{KL}(X, Y)$ como función de Y dará un resultado distinto del que se obtendría optimizando sobre X .

Estas dos funciones de error o pérdida, darán propiedades diferentes a las matrices W y B . Para entender las diferencias entre estas dos funciones, hay que observar que la función del error KL obtiene un logaritmo de una razón. Esto no suele permitir razones muy grandes entre la matriz original y la matriz aproximada. De hecho, regiones con una intensidad más baja contribuyen más a la función del error o pérdida que regiones con intensidad más alta si la razón entre ellas es suficientemente grande. El término con un logaritmo de una razón no está en la función de error de Frobenius, que sólo no permite diferencias grandes. Esto último viene bien para la aproximación de regiones con una intensidad más grande, ya que contribuye a que haya más vectores de la base concentrados en dichas regiones. Debido a la restricción para que sean no negativos, la base B^* suele ser disjunta; es decir, los distintos tipos de tiro que forman la base B^* tienen intersección vacía entre ellos, exhibiendo una descomposición más fragmentada que la obtenida con otros métodos, como por ejemplo el método de componentes principales (PCA).

3.3. Datos

Una vez se ha explicado la base teórica del estudio, vamos a introducir los datos que se van a utilizar y los resultados obtenidos. Hay que observar que tanto los datos como los resultados los hemos tomado de Miller y cols. (2014). Los datos que se han utilizado son los tiros de campo intentados y anotados con sus respectivas localizaciones en la pista de casi la mitad de los partidos de la temporada regular 2012-2013 de la NBA. Se quitan los jugadores con menos de 50 tiros de campo intentados, obteniéndose un total de 78000 tiros distribuidos entre 335 jugadores

de la NBA.

Se modelará los tiros de cada jugador con un proceso de puntos en la mitad ofensiva de la pista, un rectángulo de 35 ft por 50 ft, lo que viene a ser un rectángulo de 9,144 m por 15,24 m. A cada jugador se le identificará con $n \in \{1, \dots, N\}$, y denotamos $\mathbf{x}_n = \{x_{n1}, \dots, x_{nM_n}\}$, al conjunto de tiros del jugador n -ésimo, siendo M_n el número de tiros total del jugador n -ésimo y $x_{nm} \in [0, 35] \times [0, 50]$.

Se representará el resultado del tiro con una variable $y_{nm} \in \{0, 1\}$ para indicar si el n -ésimo jugador ha anotado el m -ésimo tiro (1) o lo ha fallado (0).

El objetivo que se pretende conseguir con estos datos es representar gráficamente los distintos tipos de hábitos de tiro que presentan los jugadores NBA.

3.3.1. Observación de la No Estacionariedad

Como un paso previo para explorar los datos, se visualizará la correlación espacial empírica entre los conjuntos de los tiros en un espacio discreto. Se divide la cancha en V partes, y se introduce X como $X_{n,v} = |\{x_{n,i} : x_{n,i} \in v\}|$, el número de tiros del jugador n en la zona v . La correlación empírica representada en la Figura 3.1 respecto a dos zonas de la pista (la zona de debajo del aro y la zona del triple marcadas con una \otimes), da una idea de la no estacionariedad de las intensidades en las distintas zonas. Esto motiva el uso de métodos adicionales, como el descrito anteriormente NMF, para introducir patrones globales que expliquen estas correlaciones entre puntos tan lejanos.

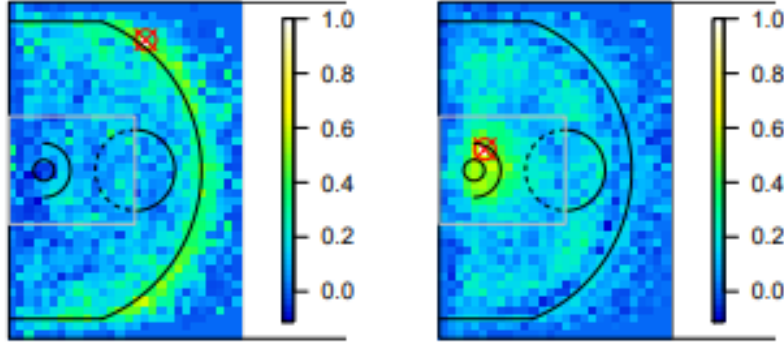


Figura 3.1: Correlación espacial empírica entre la zona marcada con una cruz y las demás. Estos datos muestran que hay patrones de correlación no estacionarios, especialmente entre los tiradores de triples y sugiere modelizar un mecanismo que resuelva en la medida de los posible la correlación global.

3.4. Idea aproximada

El método reside en la aplicación de procesos Log-Gaussianos Cox y en las matrices de factorización no negativas para obtener patrones espaciales de los datos de tiros de la NBA.

Dados los procesos de puntos de cada uno de los N jugadores, $\mathbf{x}_1, \dots, \mathbf{x}_N$, el procedimiento es el siguiente

1. Construir la matriz $X_{n,v} = \#$ tiros del jugador n en la zona v de la cancha.
2. Utilización de los procesos Log-Gaussianos Cox para construir una superficie de intensidad $\lambda_n = (\lambda_{n,1}, \dots, \lambda_{n,V})^T$ para cada jugador n la cancha fragmentada. Aquí utilizamos los procesos Log-Gaussianos Cox.
3. Obtener la matriz de datos $\Lambda = (\bar{\lambda}_1, \dots, \bar{\lambda}_N)^T$, donde $\bar{\lambda}_n$ ha sido normalizada para que $\sum \bar{\lambda}_n = 1$.
4. Encontrar las matrices B y W para algún K tal que $\Lambda \approx WB$ utilizando

las matrices de factorización no negativas, con la condición que todos los términos sean no negativos.

Esto último resulta en una base espacial B y los pesos de dicha base para cada jugador, w_n . Debido a la propiedad de superposición de los procesos de Poisson y la no negatividad de los elementos de la base y de los pesos, los vectores de la base pueden ser interpretados como funciones de subintensidad. Los pesos de cada jugador resumen muy bien los hábitos de tiro de dicho jugador en un vector \mathbb{R}_+^K .

Aunque se ha descrito un modelo continuo por simplicidad conceptual, ya se ha visto que la cancha está discretizada en V zonas cuadradas de 1 pie por 1 pie, para ganar tratabilidad computacional para ajustar las superficies de los procesos Log-Gaussianos Cox (LGCPs). Se espera que el tamaño de las zonas sea adecuado para capturar toda la variación espacial que sea interesante.

3.4.1. Ajuste de LGCPs

Para cada conjunto de puntos de cada jugador, \mathbf{x}_n , la probabilidad del proceso de puntos puede ser aproximada de forma discreta como

$$p(\mathbf{x}_n | \lambda_n(\cdot)) \approx \prod_{v=1}^V p(X_{n,v} | \Delta A \lambda_{n,v}),$$

donde $\lambda_n(\cdot)$ es la función de intensidad exacta, mientras que λ_n es la función de intensidad discreta (vector), y ΔA es el área de cada zona de la pista (que es de un pie). Esta aproximación viene de la aleatoriedad en el espacio de los procesos de Poisson, que nos permite tratar cada zona de la pista de forma independiente.

La probabilidad de tiro en cada zona es una Poisson con una intensidad uniforme $\lambda_{n,v}$.

Explícitamente, se representa el campo aleatorio Gaussiano \mathbf{z}_n , su probabilidad a posteriori es

$$p(\mathbf{z}_n|\mathbf{x}_n) \propto p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n) = \prod_{v=1}^V e^{-\lambda_{n,v}} \frac{\lambda_{n,v}^{X_{n,v}}}{X_{n,v}!} N(\mathbf{z}_n|0, K),$$

$$\lambda_n = \exp(\mathbf{z}_n + z_0),$$

donde la priori sobre \mathbf{z}_n es una Normal de media cero y con matriz de covarianzas $K_{v,u} = k(x_v, x_u)$, determinado por la ecuación 3.2.1, y z_0 es el sesgo que parametriza la razón media del proceso de Poisson. Transformando muestras de $\mathbf{z}_n|\mathbf{x}_n$ se puede construir muestras de la probabilidad a posteriori de $\lambda_n|\mathbf{x}_n, p(\lambda_n|\mathbf{x}_n)$. Para superar las correlaciones tan altas inducidas por la estructura espacial de la pista se utilizarán muestras parciales elípticas Murray y cols. (2010) para aproximar la posteriori de λ_n para cada jugador, y como consecuencia, la media a posteriori.

3.4.2. Optimización NMF

Se resuelve el problema de optimización 3.2.2 utilizando técnicas de D. D. Lee y Seung (2001) y Brunet y cols. (2004), comparando las funciones KL y Frobenius de error o pérdida para ver las principales diferencias entre los vectores base que resultan.

3.5. Resultados

A continuación, se presentan los resultados obtenidos para nuestro proceso de puntos, donde se ha utilizado LGCP con NMF con $K=10$. Este método obtiene vectores base que se corresponden a tipos de tiros que se pueden representar visualmente. Además, los pesos para la base de cada jugador da una caracterización muy precisa de sus hábitos ofensivos como se puede observar en la Figura 3.2. El peso $w_{n,k}$ puede ser interpretado como cantidad de veces que el jugador n hace un tiro del tipo k , lo que hace que nos hagamos una idea muy clara del comportamiento ofensivo del jugador.

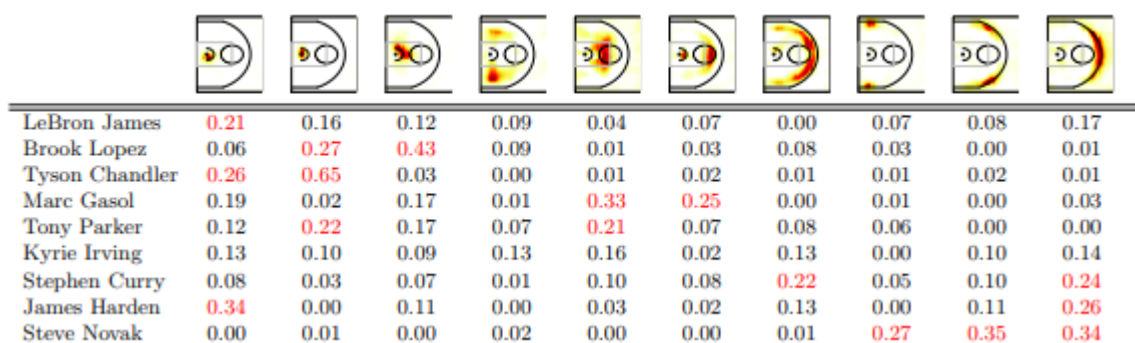


Figura 3.2: Pesos normalizados para la base de distintos jugadores NBA.

En la Figura 3.2 se pueden observar perfiles de jugadores muy distintos entre sí. Por ejemplo, se puede observar que los pesos más altos de LeBron James se producen bajo el aro, y en la zona frontal del triple. Lo que nos viene a decir que es un jugador hábil en el tiro de 3 pero con suficiente embergadura y físico como para aprovecharlo bajo el aro. En cambio, están los perfiles de los triplistas, como son Steve Novak y Stephen Curry, con pesos muy altos en todos los tiros de 3, tanto en los tiros desde las esquinas, como en las alas, como en la parte frontal. Este

tipo de gráfico viene muy bien para ver rápidamente las características ofensivas de un jugador determinado.

Empíricamente, la descomposición con NMF utilizando la función de error KL tiene una base que es más diversa respecto al espacio de la pista que la que se obtiene con la descomposición NMF y la función de error de Frobenius, en la que podemos observar que se centra mucho en la zona cercana al aro, mientras que no representa bien la zona de 2 puntos y cercana al triple. Como se puede observar en la Figura 3.3.

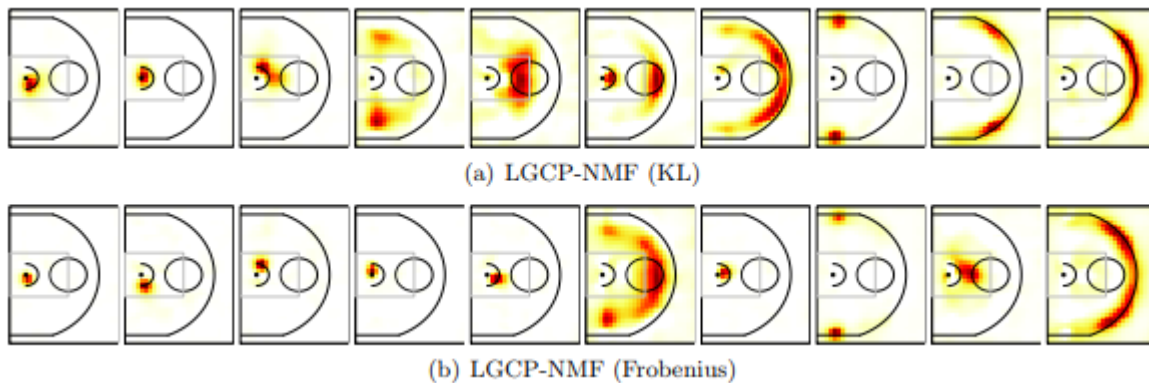


Figura 3.3: Comparación entre las bases obtenidas con el método de factorización NMF y función de error KL y el obtenido con el método de factorización NMF y función de error Frobenius.

3.6. Eficiencia del tiro

El porcentaje de tiros de campo, o la probabilidad de que un jugador enceste un tiro, son estadísticas de interés cuando se quiere saber el valor de un jugador. Sin embargo, esta estadística es uniforme en el espacio, y no contempla que hay una variación muy importante según las características en las que se ha hecho el intento. No es lo mismo tirar un triple sin presión y a falta de 5 segundos de

posesión, que intentar un triple con 2 segundos de posesión y con dos jugadores rivales defendiendo.

Aprovechando el vocabulario de los tipos de tiros que se ha introducido en el apartado anterior con los vectores base, se modelará el porcentaje de tiros de campo de cada jugador para cada tipo de tiro. Se descompondrá el porcentaje global de tiro de cada jugador en una combinación de K porcentajes en distintos tipos de tiro (que son bases) con sus correspondientes pesos. Como se puede observar, esto ofrece un resumen más completo de las habilidades ofensivas de cada jugador.

El objetivo que se pretende en los siguientes apartados es estimar la probabilidad de anotar el tiro para cada punto de la zona ofensiva de la pista y para cada jugador. Antes de comenzar con la descripción del modelo, se verán algunas definiciones necesarias para la comprensión de los siguientes apartados.

Definición 3.6.1. *Una variable aleatoria k -dimensional $X = (X_1, X_2, \dots, X_k)$ se dice que sigue una distribución multinomial de parámetros n y p_1, p_2, \dots, p_k y se denota por $X \sim \text{Mult}(n; p_1, p_2, \dots, p_k)$ si su función de densidad viene dada por la expresión:*

$$\begin{cases} \frac{n!}{x_1! x_2! \dots x_k!} p_1^{x_1} p_2^{x_2} \dots p_k^{x_k} (1 - \sum_{i=1}^k p_i)^{n - \sum_{i=1}^k x_i} & \text{con } x_i = 0, 1, \dots, n \text{ y } \sum_{i=1}^k x_i \leq n; \\ 0 & \text{en otro caso.} \end{cases}$$

Nota: Una variable aleatoria (v.a.) discreta X se dice que sigue una distribución de Bernoulli de parámetro p , $X \sim \text{Bern}(p)$, con $p \in [0, 1]$; si se verifica que $\text{rg}(X) = \{0, 1\}$ y la función de probabilidad puede describirse como:

$$P[X = x] = p^x (1 - p)^{1-x} \text{ con } x \in \{0, 1\}$$

3.6.1. Descripción del modelo

Para el jugador n , se modeliza cada tiro como

1. Tipo de tiro

$$k_{n,i} \sim \text{Mult}(\bar{w}_{n,k})$$

2. Localización

$$x_{n,i}|k_{n,i} \sim \text{Mult}(\bar{B}_{k_{n,i}})$$

3. Resultado

$$y_{n,i}|k_{n,i} \sim \text{Bern}(\text{logit}^{-1}(\beta_{n,k_{n,i}}))$$

donde $\bar{B}_k \equiv B_k / \sum_{k'} B_{k'}$ es la base normalizada, y los pesos de cada jugador, $\bar{w}_{n,k}$ están ajustados para reflejar la masa total de cada base sin normalizar. La técnica de matriz no negativa de factorización (NMF) no contrae cada vector de la base a un valor único, pero el volumen de cada vector de la base es una cantidad muy útil que corresponde a cómo de común es un tipo de tiro. Se transfiere esta información a los pesos de la siguiente manera, obteniéndose los pesos ajustados para la base

$$\bar{w}_{n,k} = w_{n,k} \sum_v B_k(v).$$

No se puede observar directamente el tipo de tiro, k , solo la localización de dicho tiro $x_{n,i}$. Omitiendo n e i para simplificar la notación, podemos computar la distribución predictiva

$$p(y|x) = \sum_{k=1}^K p(y|k)p(k|x) = \sum_{k=1}^K p(y|k) \frac{p(x|k)p(k)}{\sum_{k'} p(x|k')p(k')}.$$

La descomposición del tipo de tiro dado por B proporciona una forma natural de compartir información entre distintos tipos de tiradores para reducir la varianza de nuestras superficies estimadas. Se modelará jerárquicamente los parámetros de probabilidad de cada jugador $\beta_{n,k}$ respecto a cada tipo de tiro. Antes se definirá las distribuciones Gamma y Gamma-Inversa.

Definición 3.6.2. *Se dice que la v.a. X sigue una distribución Gamma de parámetros $a > 0$ y $\lambda > 0$ ($X \sim Ga(a, \lambda)$) si su función de densidad es:*

$$f_X(x|a, \lambda) = \frac{\lambda^a}{\Gamma(a)} x^{a-1} e^{-\lambda x}, \text{ con } x > 0.$$

Definición 3.6.3. *Se dice que la v.a. X sigue una distribución Gamma Invertida de parámetros $a > 0$ y $\lambda > 0$ ($X \sim GaI(a, \lambda)$) si su función de densidad es:*

$$f_X(x|a, \lambda) = \frac{\lambda^a}{\Gamma(a)} \frac{1}{x^{a-1}} e^{-\lambda/x}, \text{ con } x > 0.$$

Los parámetros a priori son

1. Priori global difusa

$$\beta_{0,k} \sim N(0, \sigma_0^2).$$

2. Varianza de la base

$$\sigma_k^2 \sim GaI(a, b).$$

3. Parámetros jugador/base

$$\beta_{n,k} \sim N(\beta_{0,k}, \sigma_k^2),$$

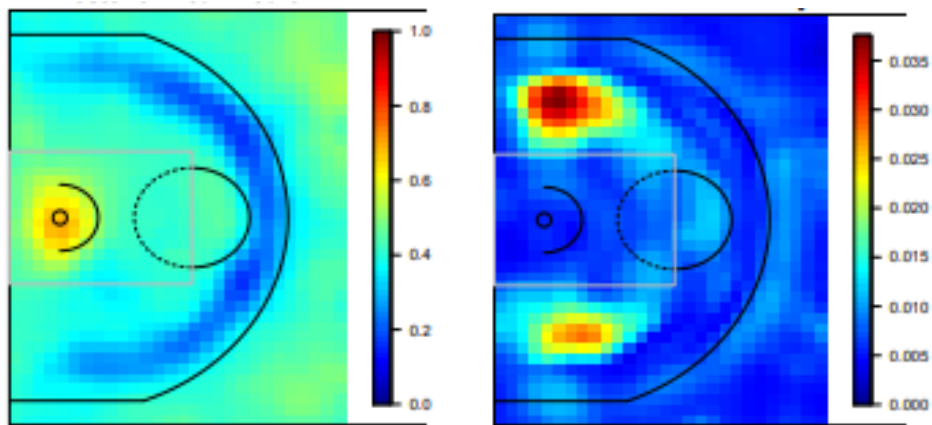
donde las medias globales, $\beta_{0,k}$, y las varianzas, σ_k^2 , vienen dadas por a prioris

difusas, σ_0^2 , y $a = b = 0,1$. El objetivo de esta estructura jerárquica de la a priori es compartir información entre los jugadores sobre un tipo de tiro concreto. Además reduce a los jugadores con tamaños de muestra pequeños a la media global. Algunas de las consecuencias de tomar esta estructura para modelar serán discutidas en la sección 3.7.

3.6.2. Inferencia

Las muestras de Gibbs se utilizan para dibujar las muestras a posteriori de los parámetros β y σ^2 . Para dibujar las muestras a posteriori de $\beta|\sigma^2, y$ se utilizan muestras elípticas para aprovechar la distribución a priori normal de β .

3.6.3. Resultados



(a) Superficie de la eficiencia global a posteriori

(b) Incertidumbre a posteriori

Figura 3.4: En (a) se puede observar la superficie de la eficiencia posteriori, en cambio en (b), se observa la incertidumbre a posteriori.

Se puede ver en la Figura 3.4a la media global de la eficiencia, que se corresponde con los parámetros $\beta_{0,k}$. En cambio, en la Figuras 3.5, 3.6, 3.7 y 3.8 se muestran algunos ejemplos de la eficiencia de algunos jugadores en las distintas zonas de la pista, lo que nos permite comparar la eficiencia de dichos jugadores en las distintas zonas. De hecho, se puede observar como jugadores como Steve Novak y Stephen Curry son muy propensos al tiro de 3, en cambio, Kyrie Irving y LeBron James son jugadores con una eficiencia más alta en la zona interior, cerca del aro. Cabe destacar también, la baja eficiencia de jugadores como Steve Novak y Kyrie Irving en los tiros *baseline jump*, que corresponde a las zonas laterales entre la zona y el triple. Stephen Curry destaca por tener una eficiencia altísima en zonas puntuales, como los triples abiertos y la zona alta de la zona.

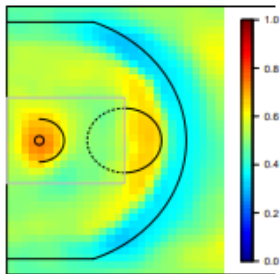


Figura 3.5: EP de LeBron James

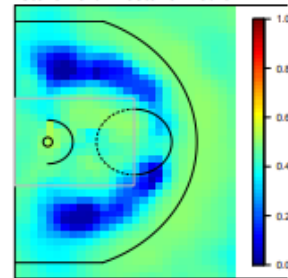


Figura 3.6: EP de Steve Novak

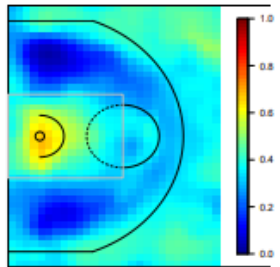


Figura 3.7: EP de Kyrie Irving

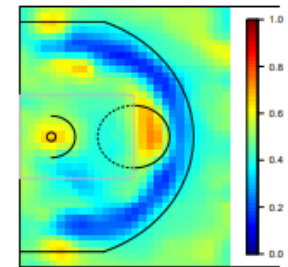


Figura 3.8: EP de Stephen Curry

3.7. Conclusiones

Se ha presentado un método que modela los procesos de puntos utilizando técnicas de matrices de factorización para ajustar superficies de intensidad. Esto proporciona un resumen muy preciso de los hábitos de tiro de los distintos jugadores y una base intuitiva de los tipos de tiros fácilmente reconocibles por los entrenadores y fans. Desde el punto de vista táctico, el análisis de los hábitos de tiro de los distintos jugadores y de la efectividad que tienen los mismos ofrece una idea rápida y concisa de cómo se comportan dichos jugadores en la pista. Posibles líneas futuras para este trabajo es ver la diferencia en los hábitos de tiro y sobre todo la eficiencia entre distintas competiciones, como temporada regular y *Play-Offs*, ya que suele haber un comportamiento muy distinto de los jugadores en las distintas competiciones.

3.7. CONCLUSIONES

4 MÉTRICAS

*“Las ganas de ganar son importantes,
pero las ganas de prepararse son vitales”*

Joe Paterno (1926-2012)

4.1. Introducción

Hay muchas formas de hacerse una idea de la eficiencia de un jugador en la pista en diversos aspectos ofensivos o defensivos. Normalmente, se utilizan las métricas, que son valores que representan el comportamiento del jugador en dichos factores. Tales valores se calculan con los resultados de ese jugador en uno o más partidos, por ejemplo, la eficiencia de anotación de un jugador es una métrica ofensiva muy importante, ya que refleja la influencia del jugador en el partido o en su equipo, la cual se explicará en las siguientes secciones. Ahora bien, en este capítulo, se compararán las métricas ofensivas tradicionales con el resultado obtenido de modelar la eficiencia en la anotación con una familia conjugada Multinomial-Dirichlet para obtener una fórmula de credibilidad (véase Pulgarín, Arias-Nicolás, y Jiménez (2017)). A continuación, se explicarán algunos conceptos clave para la comprensión de los distintos aspectos que se explicarán a lo largo del capítulo.

4.1.1. Conceptos previos

Como se ha comentado en la introducción, hay ciertos indicadores del juego que pueden ser utilizados en conjunto para medir la productividad ofensiva de un jugador en la pista. Estos indicadores son más avanzados y contienen más

4.1. INTRODUCCIÓN

información que los que aparecen en la típica tabla resumen de un partido o promedios de una temporada (véase Figura 4.1), que son más planos y menos útiles a la hora de valorar los distintos aspectos ofensivos y defensivos en un conjunto.

		Totals																Shooting		
Starters	Tm	MP	FG	FGA	3P	3PA	FT	FTA	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	FG%	3P%	FT%
LeBron James	CLE	31:17	12	17	4	8	1	1	0	10	10	8	1	0	5	2	29	.706	.500	1.000
Kevin Durant	GSW	29:57	7	13	3	8	2	2	1	5	6	5	3	1	2	3	19	.538	.375	1.000
Russell Westbrook	OKC	28:28	4	11	1	6	2	3	3	5	8	8	0	0	2	2	11	.364	.167	.667
Kyrie Irving	BOS	27:36	6	16	1	5	0	0	2	5	7	9	1	0	1	2	13	.375	.200	
Anthony Davis	NOP	16:40	6	9	0	1	0	0	1	1	2	1	0	0	1	2	12	.667	.000	
		Totals																Shooting		
Reserves	Tm	MP	FG	FGA	3P	3PA	FT	FTA	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	FG%	3P%	FT%
Paul George	OKC	26:30	6	15	4	10	0	0	1	4	5	4	1	0	0	2	16	.400	.400	
Andre Drummond	DET	17:36	7	7	0	0	0	0	1	2	3	0	0	1	1	2	14	1.000		
Bradley Beal	WAS	16:51	5	10	4	8	0	0	0	0	0	0	1	0	1	1	14	.500	.500	
Victor Oladipo	IND	15:00	3	8	1	6	0	0	0	2	2	3	3	0	0	0	7	.375	.167	
Kemba Walker	CHO	14:09	5	10	1	5	0	0	1	1	2	1	0	0	0	0	11	.500	.200	
Goran Dragic	MIA	11:28	1	3	0	1	0	0	0	4	4	1	0	0	1	2	2	.333	.000	
LaMarcus Aldridge	SAS	4:28	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	.000		
Team Totals		240	62	120	19	58	5	6	10	39	49	40	10	3	14	18	148	.517	.328	.833

Figura 4.1: Tabla con las estadísticas básicas del equipo LeBron en el *All-Star Games* de 2018

Una manera de comparar estadísticas más efectiva es ver cuántas veces tienen oportunidad de anotar puntos en el juego en cuestión, estas oportunidades son denominadas posesiones. Realmente, una posesión es todo el tiempo que un equipo posee la pelota antes de que el rival la obtenga, lo que viene siendo el control ininterrumpido del balón. Hay métricas que estiman muy bien el número de posesiones que tiene un equipo en un partido. Pero véanse las reglas que definen lo que es una posesión, y en la que se basa la fórmula que estima el número de posesiones en un partido.

Hay solo tres formas de que una posesión termine:

1. Un intento de anotación, *a field goal attempt* (denotado como FGA), cuyo rebote no es ofensivo; es decir, la defensa obtiene la pelota (si encesta también se pierde la posesión, por lo que entra en este apartado).
2. Una pérdida de balón, *a turnover* (denotado como *Tov*).
3. Algunos tiros libres, *free throws* (denotado como FTA). Hay que observar que no todos los tiros libres finalizan posesiones. La mayoría de los tiros libres vienen en pares, después de una falta en un tiro de dos, o después de la quinta falta de equipo en un mismo cuarto entre otras causas, y el primero de los tiros no termina la posesión, por lo que hay que tenerlo en cuenta. Es cierto, que los tiros libres provocados por faltas técnicas o faltas flagrantes no terminan la posesión. Por último, también observar que si la falta se produce en un tiro de tres y el jugador falla, hay tres tiros libres, de los cuales, los dos primeros no terminan la posesión. En la NBA, se ha llegado a la conclusión que el 44% de los intentos de tiro de campo corresponden al número de posesiones producidas por tiros libres (FTA).

Una vez explicadas las causas por las que termina una posesión, se verá la fórmula que estima el número de posesiones de un jugador (denotada como *Poss*) en base a estadísticas básicas. Las definiciones de las métricas que se utilizarán están tomadas de la página oficial de estadísticas de la NBA (véase NBA (27 de Abril de 2018)).

$$Poss = FGA + 0,44 \cdot FGA + Tov \tag{4.1.1}$$

Obsérvese que hay un ámbito en el que la definición de posesión se vuelve un poco inconsistente ya que hay dos definiciones diferentes usadas por varios analistas. La principal discrepancia entre estos analistas es si termina o no una posesión después de un rebote ofensivo. La jugada del equipo no termina necesariamente después de un tiro y su consiguiente rebote ofensivo, por lo que la posesión se extiende. En cambio, no ocurre lo mismo cuando se habla de las posesiones de un jugador en particular. La principal diferencia entre la posesión individual o de equipo es que, en el primer caso el rebote ofensivo empieza una nueva posesión, y en el segundo caso continúa la anterior.

Ahora, se introducirá el indicador (USG%) que representa la estimación del porcentaje de posesiones de equipo que utiliza un jugador mientras está en la pista. Obsérvese que los indicadores USG% son valores exactos y no estimados, los cuales se obtienen de la página oficial de estadísticas NBA.

Con fórmula de la estimación del número de posesiones descrita en 4.1.1 y utilizando el indicador anterior, se introducirá la métrica que nos estima el número de posesiones totales del equipo solamente cuando el jugador está en la pista:

$$Poss^{Tm} = 100 \cdot (Poss/Us\%). \quad (4.1.2)$$

4.1.2. Rating Ofensivo

La *Eficiencia Ofensiva* (en inglés *Offensive rating*) es la cantidad de puntos anotador por un jugador cada 100 posesiones:

$$OEff = 100 \cdot (Pts/Poss). \quad (4.1.3)$$

Hay que observar que el concepto de rating ofensivo puede tener dos significados muy diferentes dependiendo de qué fuente lo proporcione. Según la página oficial de la NBA (NBA.com), el rating ofensivo es el reflejo de los puntos que ha anotado el equipo por cada 100 posesiones mientras el jugador está en pista; es decir,

$$OffRtg = 100 \cdot (Poss/Usg \%). \quad (4.1.4)$$

Combinando 4.1.2 y 4.1.4, se puede calcular:

$$Pts^{Tm} = OffRtg \cdot (Poss/Usg \%). \quad (4.1.5)$$

En cambio, el rating ofensivo publicado en Basketball Reference (denotado como ORtg) (véase BR (5 de Mayo de 2018)) es una fórmula muy compleja que estima los puntos producidos por el jugador, mediante los puntos anotados, las asistencias y los rebotes ofensivos, por cada 100 posesiones de dicho jugador. Este indicador fue creado por Dean Oliver (véase Kubatko, Oliver, Pelton, y Rosenbaum (2007)).

En nuestro caso, no se discrepa en el hecho de que la métrica *OffRtg* refleja

parte del impacto de la eficiencia anotadora de un jugador en el partido, pero esto no quiere decir que se apoye la idea de que el indicador $OEff$ descrito anteriormente, deba tener un cierto peso en el cálculo de dicha métrica, y sería importante medir “cómo de creíble” es dicho indicador.

4.1.3. Teoría de la credibilidad

La teoría de la credibilidad se originó y desarrolló para Ciencias Esconómicas y Actuariales para determinar el riesgo en ciertos eventos, combinando la experiencia individual y el conocimiento colectivo que se tiene a priori. Las fórmulas de credibilidad derivan del Teorema de Bayes, y más adelante se utilizaron técnicas Bayesianas más avanzadas, que se han seguido utilizando hasta ahora.

En este caso, se aboga por una fórmula de credibilidad simple para la eficiencia anotadora de un jugador, combinando la información individual del mismo y la del equipo. Para ello, se utilizará el indicador $OEff$ para reflejar la eficiencia “directa” del jugador en el partido y $OEff^{Tm}$ para reflejar la eficiencia “residual”; es decir, que aunque el jugador en cuestión no anote, cómo afecta al juego o la eficiencia de sus compañeros.

Definición 4.1.1. *Se define la fórmula de credibilidad para la eficiencia anotadora de un jugador como:*

$$SEff = C \cdot OEff + (1 - C) \cdot OEff^{Tm}, \quad (4.1.6)$$

donde $C \in [0, 1]$ se denomina factor de credibilidad y es una función proporcional al $Usg\%$ (muchos autores defienden que $Usg\%$ está muy relacionado con el $OEff$, por lo que es lógico que el factor de credibilidad sea proporcional al indicador $Usg\%$).

4.2. Método

El objetivo en este capítulo es estudiar la variable categórica X determinando el número de puntos ($s = 0, 1, 2, 3$) anotados por un jugador en una posesión dada. Se supone que los parámetros de la distribución son $\mathbf{p} = (p_0, p_1, p_2, p_3)$, con $\sum_{s=0}^3 p_s = 1$. También se tiene que $p_s = P[X = s]$, a priori determina lo propenso que es el equipo a anotar s puntos en una posesión ofensiva cuando el jugador está en la cancha.

Se identificará X con una variable aleatoria multinomial (se justifica el uso de la multinomial en Parker (2010)) explicada en 3.6.1:

$$\mathbf{X} = (X_0, X_1, X_2, X_3) \sim Mult(\mathbf{p}),$$

donde X_s denota el número de posesiones en las que el jugador anota s puntos. Nótese que X es una variable aleatoria que especifica cada resultado individual, mientras que la distribución multinomial \mathbf{X} especifica el número de resultados en cada una de las categorías.

Sea $n = Poss$ el número total de posesiones individuales utilizada por el jugador en el partido y se considera la muestra $S = \{s_1, \dots, s_n\}$ consistente en los puntos anotados por dicho jugador en cada una de las n posesiones, y por lo tanto, $\bar{S} = OEff/100$. Antes de seguir, se describirá la distribución de Dirichlet que se utilizará más adelante:

Definición 4.2.1. Sea $\mathbf{X} = (X_1, \dots, X_n)$, se dice que sigue una distribución Dirichlet con parámetros $(\alpha_1, \dots, \alpha_n)$ con $\alpha_i > 0 \forall i$ (denotado como $\mathbf{X} \sim Di(\alpha_1, \dots, \alpha_n)$) si su función de densidad es:

$$f(x_1, \dots, x_n | \alpha_1, \dots, \alpha_n) = \frac{\Gamma(\sum_{i=1}^n \alpha_i)}{\prod_{i=1}^n \Gamma(\alpha_i)} x_1^{\alpha_1-1} \dots x_n^{\alpha_n-1}, \text{ con } 0 < x_i < 1 \forall i \text{ y } \sum_{i=1}^n x_i = 1.$$

La función de verosimilitud basada en la muestra $S = \{s_1, \dots, s_n\}$ viene dada por:

$$L(\mathbf{p}|S) = \prod_{k=1}^n p_{s_k} = p_0^{n_0} p_1^{n_1} p_2^{n_2} p_3^{n_3}, \quad (4.2.1)$$

donde n_s indica el número de posesiones en las cuales el jugador anota s puntos, por lo cual, $\sum_{s=0}^3 n_s = n$ y se denota $\mathbf{n} = (n_0, \dots, n_3)$. Recuérdese que la función de probabilidad de la multinomial es proporcional a 4.2.1:

$$P[\mathbf{X} = \mathbf{n}] = \begin{cases} \frac{n!(p_0^{n_0} p_1^{n_1} p_2^{n_2} p_3^{n_3})}{n_0! n_1! n_2! n_3!}, & \text{cuando } n_0 + n_1 + n_2 + n_3 = n; \\ 0, & \text{en otro caso.} \end{cases}$$

La aproximación de Bayes se basa en el conocimiento que hay en las distribuciones a priori de las muestras para caracterizar dichas probabilidades. El apoyo empírico para la distribución de Dirichlet (véase 4.2.1) es que da un mejor ajuste a la distribución de la frecuencia anotadora que si asumimos que todas las distribuciones individuales tienen la misma distribución (véase T. C. Lee, Judge, y Zellner (1968)). Por tanto, se utilizará la distribución de Dirichlet para el parámetro \mathbf{p} , de manera que:

$$\mathbf{p} \sim \text{Dir}(\mathbf{a}), \text{ i.e. } \Pi(\mathbf{p}) = \frac{\Gamma(a)(p_0^{a_0-1} p_1^{a_1-1} p_2^{a_2-1} p_3^{a_3-1})}{\Gamma(a_0)\Gamma(a_1)\Gamma(a_2)\Gamma(a_3)},$$

y $E_{\mathbf{p}}[p_s] = a_s/a$, donde $\mathbf{a} = (a_0, a_1, a_2, a_3)$ son los parámetros de la distribución a priori (hay que observar que son distintos que los parámetros \mathbf{p} , que son desconocidos, a diferencia de \mathbf{a} que son conocidos) y se nota $a = a_0 + a_1 + a_2 + a_3$.

Aplicando el teorema de Bayes para variables discretas explicado anteriormente, se obtiene una función de distribución a posteriori:

$$\Pi(\mathbf{p}|S) = \frac{L(\mathbf{p}|S)\Pi(\mathbf{p})}{E[L(\mathbf{p}|S)]} = \frac{\Gamma(a+n)(p_0^{a_0+n_0-1} p_1^{a_1+n_1-1} p_2^{a_2+n_2-1} p_3^{a_3+n_3-1})}{\Gamma(a_0+n_0)\Gamma(a_1+n_1)\Gamma(a_2+n_2)\Gamma(a_3+n_3)},$$

por tanto, $\mathbf{p} \propto \text{Diri}(\mathbf{a} + \mathbf{n})$, y entonces se tiene que $E_{\mathbf{p}|S}[p_s] = (a_s + n_s)/(a + n)$.

Ahora se define los principios para el cálculo de la eficiencia anotadora de cada jugador. La función de pérdida es una función $l : \mathbb{R}^2 \rightarrow \mathbb{R}$ que atribuye a cada par (s, x) el error cometido por un jugador en una posesión cuando se “espera” que anote x y realmente anota s .

Para calcular la eficiencia anotadora de los distintos jugadores, hay que obtener $x(\mathbf{p})$ que es el valor que minimiza la esperanza de la función pérdida $l : \mathbb{R}^2 \rightarrow \mathbb{R}$; es decir, $x(\mathbf{p})$ es el valor que hace mínimo a $E_X[l(s, x)]$.

Si se considera la función pérdida $l(s, x) = (s - x)^2$, entonces:

$$\frac{\partial}{\partial x} E_X[(s - x)^2] = E_X\left[\frac{\partial}{\partial x}(s - x)^2\right] = -2E_X[s] + 2x, \quad (4.2.2)$$

se calcula también la segunda derivada:

$$\frac{\partial^2}{\partial x^2} E_X[(s-x)^2] = E_X\left[\frac{\partial^2}{\partial x^2}(s-x)^2\right] = 2. \quad (4.2.3)$$

Como se puede observar, como en la ecuación 4.2.3 se obtiene que la segunda derivada es $2 > 0$, el valor que hace cero la ecuación 4.2.2 es un mínimo, y entonces, se obtiene que $x(\mathbf{p}) = E_X[s] = p_1 + 2p_2 + 3p_3$.

4.3. Resultados

En esta sección, se aplicará el método explicado anteriormente para calcular la eficiencia anotadora de un jugador, teniéndose una muestra S que será el valor $SEff$. Para ello, se busca minimizar la función pérdida $E_{\mathbf{p}|S}[l(x(\mathbf{p}), \frac{SEff}{100})]$.

Se procede como en el apartado anterior calculando los valores que minimizan la función pérdida y aplicando que la distribución a posteriori de \mathbf{p} es una Dirichlet, se obtiene:

$$\begin{aligned} \frac{SEff}{100} &= E_{\mathbf{p}|S}[x(\mathbf{p})] = \frac{(n_1 + a_1) + 2(n_2 + a_2) + 3(n_3 + a_3)}{n + a} = \\ &= \left(\frac{n}{n + a}\right) \left(\frac{n_1 + 2n_2 + 3n_3}{n}\right) + \left(\frac{a}{n + a}\right) \left(\frac{a_1 + 2a_2 + 3a_3}{a}\right) = \\ &= \left(\frac{Poss}{Poss + a}\right) \left(\frac{Pts}{Poss}\right) + \left(1 - \frac{Poss}{Poss + a}\right) \left(\frac{a_1 + 2a_2 + 3a_3}{a}\right), \end{aligned}$$

y aplicando la fórmula de la credibilidad (ecuación 4.1.6), se obtiene que el factor de credibilidad es:

$$C = \frac{Poss}{Poss + a}, \quad (4.3.1)$$

y que el conocimiento colectivo es:

$$OEff^{Tm} = 100 \cdot \frac{(a_1 + 2a_2 + 3a_3)}{a}. \quad (4.3.2)$$

4.3.1. Hiperparámetros

Es cierto que cuanto mayor es la muestra, mejor. De acuerdo con este principio, se puede entender a como la *precisión*, de manera que cuando a_s es suficientemente grande respecto a a , entonces a_s es cercano a p_s . En caso contrario, si a_s es pequeño respecto a a , entonces no tiene porqué ser cercano a p_s .

Teniendo en cuenta la ecuación 4.3.1 es una función creciente en función de $Usg\% = 100 \cdot \frac{Poss}{Poss^{Tm}}$, se tiene que:

- $C = 1 \Leftrightarrow (Poss^{Tm} - Poss) = 0$ y $Poss > 0 \Leftrightarrow a = 0$ y $Poss > 0$
- $C = 0 \Leftrightarrow (Poss^{Tm} - Poss) > 0$ y $Poss = 0 \Leftrightarrow a > 0$ y $Poss = 0$

Entonces, la aproximación de Naive Bayes puede afirmar que:

$$a_s = \lambda(m_s - n_s), \text{ con } 0 < \lambda < 1, \quad (4.3.3)$$

donde m_s denota el número de posesiones ofensivas en las que el equipo anota s -puntos cuando el jugador está en la pista. Además se puede obtener de la ecuación 4.3.2 y de la fórmula anterior, se tiene que el *conocimiento colectivo* es:

$$OEff^{Tm} = 100 \cdot \left(\frac{Pts^{Tm} - Pts}{Poss^{Tm} - Poss} \right), \quad (4.3.4)$$

la cual, no depende de λ .

Se puede observar que si $\lambda \rightarrow 0$, entonces $SEff = OEff$, en cambio, si $\lambda \rightarrow 1$, se tiene que $SEff = OffRtg$. De hecho, λ puede entenderse como un factor que determina cuánto más cerca se encuentra $SEff$ de $OEff$ o de $OffRtg$. Se utilizará una distribución Uniforme $U(0, 1)$ para λ , cuya media es de $\bar{\lambda} = \frac{1}{2}$ y se puede utilizar para estimar λ .

Utilizando las ecuaciones 4.3.1, 4.3.3 y 4.3.4, se obtiene que el factor de credibilidad es:

$$C = 2 \cdot \frac{Poss}{Poss + Poss^{Tm}}. \quad (4.3.5)$$

Finalmente, utilizando todo lo anterior se calcula $SEff$ y se obtiene que:

$$SEff = 100 \cdot \frac{(Pts + Pts^{Tm})}{(Poss + Poss^{Tm})}. \quad (4.3.6)$$

4.4. Discusión

En este apartado se comparará el modelo que se ha propuesto durante el capítulo con la métrica habitual para medir el valor ofensivo de los distintos jugadores, de forma que la nueva métrica puede utilizarse para evaluar la efectividad anotadora de cada jugador teniendo en cuenta las oportunidades que dispone el jugador en cuestión. Los resultados que se verán son los resultados descritos en Pulgarín y cols. (2017).

Hay que observar que se van a tener en cuenta jugadores que han jugado más de 55 partidos durante la temporada regular y que han disputado más de 20 minutos de media por partido. También deben tener un factor de credibilidad mayor que 0.3, ya que el factor de credibilidad mide cómo es de activo el jugador en cuestión. Los 25 jugadores con mayor $SEff$ y las métricas básicas se pueden ver a continuación (véase Fig. 4.2).

▲	Player	◊	Tm	◊	Poss	◊	PossTm	◊	Pts	◊	PtsTm	◊	C	◊	OffRtg	◊	SEff	◊	Dif	◊
1	Stephen Curry		GSW		25.7		80.3		30.1		93.7		0.48		116.7		116.8		0.1	
2	Klay Thompson		GSW		20.2		76.5		22.1		88.3		0.42		115.4		114.2		-1.2	
3	Draymond Green		GSW		15.1		81.2		14.0		94.5		0.31		116.4		112.7		-3.7	
4	J.J. Redick		LAC		13.9		62.1		16.3		69.2		0.37		111.4		112.5		1.1	
5	Kevin Durant		OKC		25.7		84.3		28.2		95.6		0.47		113.4		112.5		-0.9	
6	Kawhi Leonard		SAS		18.6		72.1		21.2		78.7		0.41		109.2		110.1		0.9	
7	LeBron James		CLE		24.8		79.7		25.3		89.6		0.47		112.4		110.0		-2.4	
8	Chris Paul		LAC		19.6		72.9		19.5		81.4		0.42		111.7		109.1		-2.6	
9	Enes Kanter		OKC		11.6		49.6		12.7		54.1		0.38		109.1		109.2		0.1	
10	Jonas Valanciunas		TOR		11.9		57.5		12.8		62.7		0.34		109.0		108.8		-0.2	
11	Kevin Love		CLE		16.3		69.4		16.0		77.0		0.38		111.0		108.5		-2.5	
12	LaMarcus Aldridge		SAS		17.2		66.2		18.0		72.2		0.41		109.1		108.2		-0.9	
13	Russell Westbrook		OKC		25.6		81.8		23.5		92.4		0.48		113.0		107.9		-5.1	
14	Dirk Nowitzki		DAL		17.5		68.6		18.3		73.9		0.41		107.7		107.1		-0.6	
15	Patrick Patterson		TOR		7.2		37.3		6.9		40.7		0.32		109.1		107.0		-2.1	
16	Kyle Lowry		TOR		21.3		81.9		21.2		88.9		0.41		108.5		106.7		-1.8	
17	Damian Lillard		POR		25.6		82.6		25.1		89.3		0.47		108.1		105.7		-2.4	
18	Terrence Ross		TOR		9.6		52.5		9.9		55.8		0.31		106.3		105.8		-0.5	
19	James Harden		HOU		28.8		88.6		29.0		95.0		0.49		107.2		105.6		-1.6	
20	Jae Crowder		BOS		13.7		74.5		14.2		78.9		0.31		105.9		105.6		-0.3	
21	Avery Bradley		BOS		15.5		77.5		15.2		82.9		0.33		107.0		105.5		-1.5	
22	Karl-Anthony Towns		MIN		17.7		71.7		18.3		76.0		0.40		106.0		105.5		-0.5	
23	DeMar DeRozan		TOR		23.6		79.5		23.5		85.1		0.46		107.0		105.3		-1.7	
24	Hassan Whiteside		MIA		13.2		64.4		14.2		67.4		0.34		104.7		105.2		0.5	

Figura 4.2: Tabla con las métricas del Top25 en la métrica $SEff$ durante la temporada regular en la NBA de la temporada 2015/2016

En el top se encuentra Stephen Curry (Golden State Warriors), que ha sido nombrado el MVP de la temporada 2015/2016 con una $SEff = 116.8$ que apenas

se diferencia en 0.1 de su $OffRtg$.

Antes de seguir explicando los resultados de la tabla, hay que hacer algunas observaciones, como por ejemplo que se denominan “Big-Three” a los tres jugadores all-star líderes de la conferencia Este: Kyrie Irving, LeBron James and Kevin Love en los Cleveland Cavaliers (CLE) y a los 3 jugadores líderes de los Golden State Warriors en la conferencia Oeste: Stephen Curry, Klay Thompson y Draymond Green. Obsérvese que el record de 73 victorias y 9 derrotas en temporada regular coincide con el hecho de que los “Big-Three” de Golden State estén los tres primeros respecto a la métrica $OffRtg$, en cambio, respecto a $SEff$ hay una pequeña diferencia, y es que aunque mantengan el podio en la métrica $SEff$, puede observarse que la diferencia de Draymond Green entre $OffRtg$ y $SEff$ es de -3.6, esto es provocado porque tiene un factor de credibilidad bajo ($C = 0.31$), lo que nos indica que utiliza pocas posesiones ofensivas. Hay que observar que aunque Green no es un anotador nato (14 Pts por partido), sigue en el podio (tercera posición) respecto a $SEff = 112.8$ cuando está en la cancha.

Veamos ahora a los jugadores de los Cleveland Cavaliers, Kyrie Irving no aparece en la tabla debido a que ha jugado menos de 55 partidos (53 para ser más concretos), debido a una lesión de larga duración en la temporada anterior. Aún así, Kyrie Irving tiene un factor de credibilidad de $C = 0.54$ y la fórmula explicada en este capítulo reduce su $OffRtg = 110.4$ en -3.3 para pasar a $SEff = 107.1$.

Se puede observar que hay jugadores que están sobrevalorados en el rating ofensivo y otros infravalorados. Por ejemplo, Russell Westbrook (OKC) es el jugador que presenta una variación negativa entre $OffRtg$ y $SEff$ (-5.2), también presenta un factor de credibilidad alto $C = 0.48$, por lo que esta sobrevaloración nos

indica que este jugador no produce puntos acorde con la cantidad de posesiones que consume.

El jugador más infravalorado según la fórmula desarrollada $SEff$ es J.J. Redick (LAC) que pasa de tener $OffRtg = 111.5$ a $SEff = 112.6$ y Kawhi Leonard (SAS), que pasa de un $OffRtg = 109.2$ a $SEff = 110.2$. Esto nos indica que su producción de puntos por posesión es excelente (Leonard con un factor de credibilidad de $C = 0.41$ y Redick con $C = 0.37$).

A continuación, se verá una tabla (véase Fig. 4.3) como la que hemos visto anteriormente, pero con los datos de los Playoffs de 2016, para contrastar los comportamientos de los jugadores en temporada regular y en Playoffs.

Ahora se toman los jugadores que superan los 20 minutos jugados de media por partido, un factor de credibilidad de $C = 0.3$ y un total de partidos jugados mayor que 9. En esta tabla también se puede observar que Kawhi Leonard es uno de los jugadores más infravalorados con un $OffRtg = 107.8$ y un $SEff = 108.3$ y también se vuelve a repetir que Russell Westbrook es el jugador más sobrevalorado con un $OffRtg = 112.2$ y un $SEff = 106.3$. Las explicaciones son las mismas que las ofrecidas en la temporada regular. Mientras Kawhi Leonard tiene una efectividad anotadora excelente, Westbrook no anota acorde al número de posesiones que utiliza.

No es difícil deducir que los Cleveland Cavaliers tuvieron un rendimiento bueno en los Playoffs, de hecho fueron los campeones de la NBA 2015/2016, y se puede observar viendo el Top 3 del ranking de los Playoffs (véase Fig. 4.3) donde se encuentra el “Big-Three” de Cleveland. En cambio, los jugadores que aparecen en

4.4. DISCUSIÓN

▲	Player	Tm	Poss	PossTm	Pts	PtsTm	C	OffRtg	SEff	Dif
1	Kevin Love	CLE	15.1	68.0	14.7	80.1	0.36	117.8	114.1	-3.7
2	Kyrie Irving	CLE	24.2	80.7	25.2	93.4	0.46	115.7	113.1	-2.6
3	LeBron James	CLE	26.1	85.3	26.3	97.6	0.47	114.4	111.2	-3.2
4	Kawhi Leonard	SAS	20.4	73.4	22.5	79.1	0.43	107.8	108.3	0.5
5	Draymond Green	GSW	16.5	87.3	15.4	96.8	0.32	110.9	108.1	-2.8
6	Klay Thompson	GSW	22.7	81.4	24.3	87.8	0.44	107.9	107.7	-0.2
7	Shaun Livingston	GSW	8.9	49.4	8.2	54.4	0.31	110.1	107.4	-2.7
8	LaMarcus Aldridge	SAS	20.0	72.5	21.9	77.1	0.43	106.3	107.0	0.7
9	Stephen Curry	GSW	25.0	79.6	25.1	86.3	0.48	108.4	106.5	-1.9
10	Russell Westbrook	OKC	29.5	87.5	26.0	98.2	0.50	112.2	106.2	-6.0
11	Kevin Durant	OKC	29.8	93.4	28.4	102.5	0.48	109.7	106.2	-3.5
12	CJ McCollum	POR	21.5	93.5	20.5	99.7	0.37	106.6	104.5	-2.1
13	Al-Farouq Aminu	POR	14.6	79.3	14.6	82.8	0.31	104.4	103.7	-0.7
14	Damian Lillard	POR	28.1	94.9	26.5	99.0	0.46	104.3	102.0	-2.3
15	Tony Parker	SAS	12.2	56.0	10.4	59.1	0.36	105.5	101.9	-3.6
16	Dwyane Wade	MIA	22.8	71.0	21.4	74.0	0.49	104.2	101.7	-2.5
17	Joe Johnson	MIA	13.3	72.7	12.1	75.3	0.31	103.6	101.6	-2.0
18	Kyle Lowry	TOR	22.0	82.7	19.1	85.4	0.42	103.3	99.8	-3.5
19	Jonas Valanciunas	TOR	13.0	60.5	13.8	59.5	0.35	98.3	99.7	1.4
20	Maurice Harkless	POR	12.0	57.7	11.0	58.2	0.34	100.9	99.3	-1.6
21	Jeff Teague	ATL	16.9	63.5	14.5	65.0	0.42	102.4	98.9	-3.5
22	Kent Bazemore	ATL	14.1	73.8	11.9	74.4	0.32	100.8	98.2	-2.6
23	Goran Dragic	MIA	18.3	71.2	16.5	71.1	0.41	99.9	97.9	-2.0
24	Al Horford	ATL	14.0	72.5	13.4	71.0	0.32	97.9	97.6	-0.3

Figura 4.3: Tabla con las métricas del Top25 en la métrica $SEff$ durante los Plaoffs de la NBA en 2016

el Top 3 del ranking de la temporada regular, Stephen Curry, Klay Thompson y Draymond Green, tienen ahora posiciones más lejanas como son la 9^a, 6^a y 5^a respectivamente. De aquí llama mucho la atención la bajada respecto a $SEff$ de Stephen Curry, que hace en la temporada regular $SEff = 116.8$, mientras que en Playoffs obtiene $SEff = 106.5$.

Aquí también se podría estudiar a qué jugadores les afecta más la presión de jugar las rondas clasificatorias de los Playoffs y que tienen una bajada de rendimiento considerable respecto al comportamiento medio de la temporada regular.

Por último, se verá gráficamente una comparación entre $OffRtg$ y $SEff$ de dichos jugadores durante la temporada regular NBA 2015/2016 (véase Fig. 4.4) y también otra gráfica igual pero en Playoffs (véase Fig. 4.5).

En estas gráficas se puede ver visualmente lo que hemos comentado en las tablas.

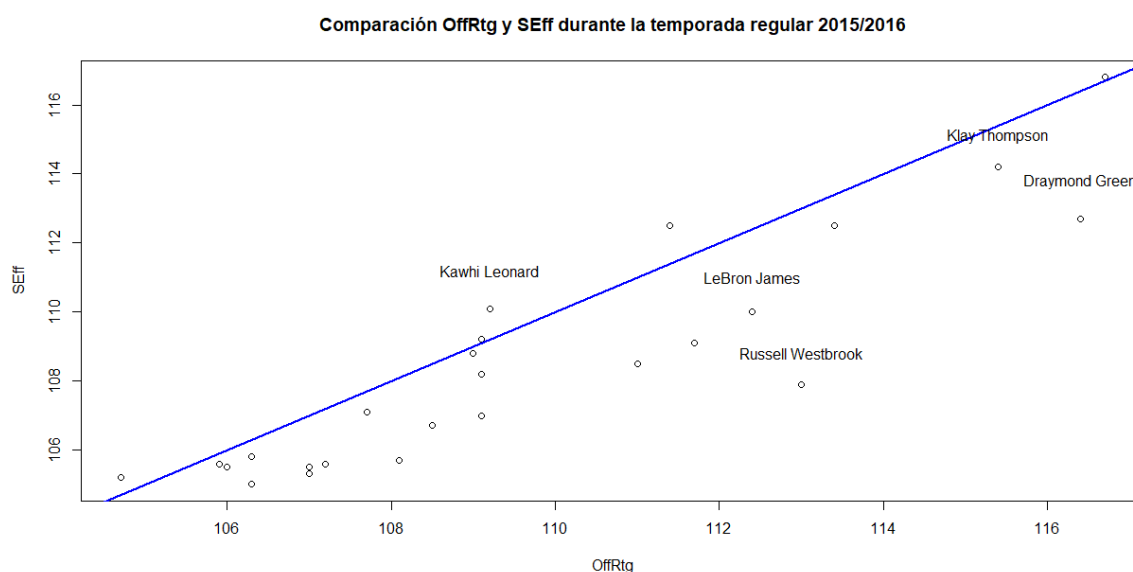


Figura 4.4: Comparación OffRtg y SEff durante la temporada regular 2015/2016

Una vez se ha introducido el concepto de *Scoring Efficiency*, posibles líneas futuras sería incorporar métricas que describan el comportamiento de ciertos grupos de jugadores, o de como esos grupos aportan al equipo más o menos rendimiento.

Incluso, se podría observar que jugadores tienen mejor compenetración en la

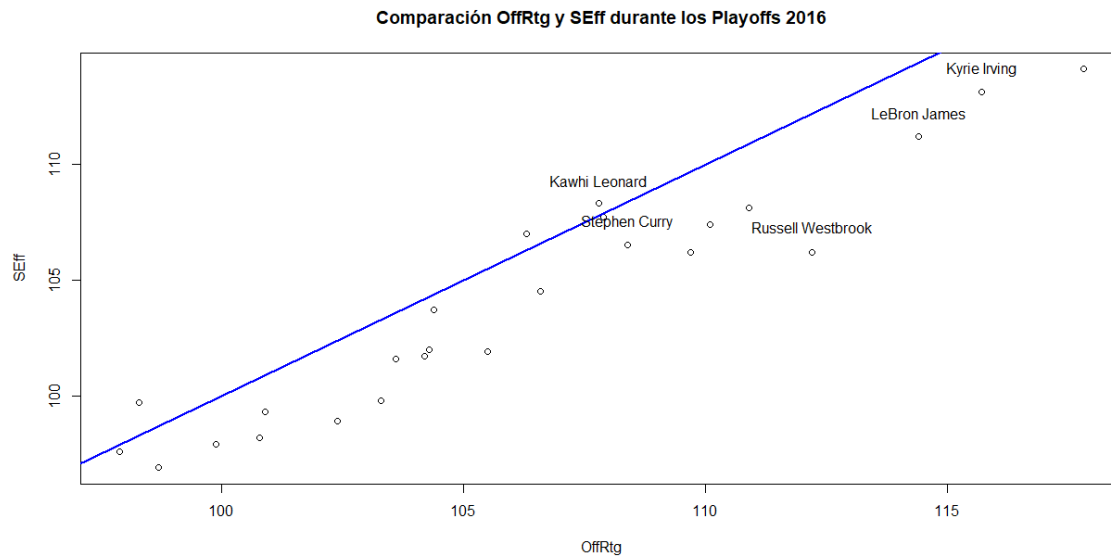


Figura 4.5: Comparación OffRtg y SEff durante los Playoffs 2016

cancha, pudiéndose optimizar el rendimiento del equipo teniendo en cuenta dicho comportamiento en grupo y podría ser clave en la toma de decisión del entrenador en los cambios.

Referencias

- BR. (5 de Mayo de 2018). <http://www.basketballreference.com/>.
- Brunet, Jean-Philippe, Tamayo, Pablo, Golub, R. T., ... P., J. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(12), 4164–4169.
- Dower, C., Rafehi, A., Weber, J., y Mohamad, R. (2018). An enhanced metric of injury risk utilizing artificial intelligence. *MIT Sloan: Sports Analytics Conference*.
- Johnston, K., Wattie, N., Schorer, J., y Baker, J. (2018). Talent identification in sport: A systematic review. *Sports Medicine*, 48, 97–109.
- Kempe, M., Grunz, A., y Memmert, D. (2015). Detecting tactical patterns in basketball: Comparison of merge self-organising maps and dynamic controlled neural networks. *European Journal of Sport Science*, 15, 249–255.
- Kubatko, J., Oliver, D., Pelton, K., y Rosenbaum, D. T. (2007). A starting point for analyzing basketball statistics. *Journal of Quantitative Analysis in Sports*, 3(3).
- Larranaga, P., Inza, I., y Moujahid, A. (2018, 06). Tema 8. redes neuronales.
- Lee, D. D., y Seung, H. S. (2001). Algorithms for non-negative matrix factorization. En T. K. Leen, T. G. Dietterich, y V. Tresp (Eds.), *Advances in neural information processing systems 13* (pp. 556–562). MIT Press.
- Lee, T. C., Judge, G., y Zellner, A. (1968). Maximum likelihood and bayesian estimation of transition probabilities. *Journal of the American Statistical Association*, 63, 1162–1179.
- McQueen, A., Wiens, J., y Gutttag, J. (2014). Automatically recognizing on-ball screens. *MIT Sloan: Sports Analytics Conference*.
- Miller, A., Bornn, L., Adams, R., y Goldsberry, K. (2014). Factorized point process intensities: A spatial analysis of professional. En *Proceedings of the 31st international conference on international conference on machine learning - volume 32* (pp. I-235–I-243). JMLR.org. Descargado de <http://dl.acm.org/citation.cfm?id=3044805.3044833>
- Murray, Iain, Adams, P., R., MacKay, y J.C., D. (2010). Elliptical slice sampling. *Journal of Machine Learning Research*, 9, 541–548.
- NBA. (27 de Abril de 2018). <http://stats.nba.com/help/glossary/>.
- Parker, R. J. (2010). *Modeling basketball's point per possession with application to predicting the outcome of college basketball games* (B.S. Thesis). College of Charleston.
- Pulgarín, A. A., Arias-Nicolás, J. P., y Jiménez, H. V. (2017). On the credibility

- of basketball scoring efficiency. *Electronic Journal of Applied Statistical Analysis*, 10(3).
- Rajiv Shah. (22 de Agosto de 2013). <https://github.com/rajshah4/BasketballData/>.
- Roseblatt. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms* (1.^a ed.). Spartan Books.
- Rumelhart, D. E., McClelland, J. L., y the PDP reserarch group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. volume i.* (1.^a ed.). MIT Press.
- Vidal, R., Bruna, J., Giryes, R., y Soatto, S. (2017). Mathematics of deep learning. *CoRR*, abs/1712.04741. Descargado de <http://arxiv.org/abs/1712.04741>
- Ward, P., Tankovich, M., Ramsden, J. S., Drust, B., y Bornn, L. (2018). Volume and intensity are important training related factors in injury incidence in american football athletes. *MIT Sloan: Sports Analytics Conference*.
- Wu, S., y Bornn, L. (2018). Modeling offensive player movement in professional basketball. *The American Statistician*, 72(1), 72-79. Descargado de <https://doi.org/10.1080/00031305.2017.1395365> doi: 10.1080/00031305.2017.1395365