Aprendizaje y toma de decisiones bajo incertidumbre

Francisco Robledo Relaño

Trabajo de fin de máster en Modelización e Investigación Matemática, Estadística y Computación
Universidad de Zaragoza

Director del trabajo: Urtzi Ayesta Morate
Ponente: José Tomás Alcalá Nalvaiz
7 de julio de 2020

Prólogo

"Of all the forms of machine learning, reinforcement learning is the closest to the kind of learning humans and other animals do"

Sutton, barto (2018)

De los paradigmas existentes en *Machine Learning*, vamos a tratar de aquel que representa la sencilla y potente idea de un sistema de aprendizaje que "quiere" algo, que adapta su comportamiento de manera que es capaz de maximizar una determinada "señal" proveniente de su entorno.

Reinforcement Learnignes una de las áreas que ha venido recibiendo más atención por parte de los investigadores en los campos de "machine learning", inteligencia artificial o redes neuronales, desarrollado en disciplinas tan diversas como psicología, teoría de control, inteligencia artificial y neurociencia. Su avance viene impulsado no solo por la creciente potencia de cálculo de los ordenadores actuales, sino por los desarrollos tan importantes que se están realizando en los aspectos de teoría y algorítmica. De todo ello trataremos en este trabajo.

Trabajo que no hubiera sido posible sin la guía y tutela del Doctor Ayesta, que ha dirigido la tesis a distancia, a causa de la pandemia, con la sobrecarga de trabajo que ello ha conllevado.

Quiero agradecer la oportunidad que nos ofreció el doctor Gorria, director en Bilbao del máster a que este trabajo pertenece, de poder colaborar con la universidad de Toulouse. Igualmente, al doctor Alcalá de la universidad de Zaragoza y a todo el cuadro de profesores del máster que nos han conducido en materias tan diversas y apasionantes.

No puedo menos de citar a mis antiguos profesores, los doctores Alexander Knebe y Gustavo Yepes de la Universidad Autónoma de Madrid, cuyo apoyo y confianza me han permitido adentrarme en un campo de tanto interés y futuro como Reinforcement Learning.

Finalmente, mi más sincero agradecimiento a mi familia por su continuo apoyo y paciencia.

Resumen

Reinforcement Learning is one of the main fields of Machine Learning, alongside Supervised and Unsupervised Learning. Unlike in the latter two cases, where the objective is the classification of the data from an already labeled sample (Supervised Learning) or from the data structure itself (Unsupervised Learning), in Reinforcement Learning the 'agent', the algorithm that carries out the learning process, learns through successive interactions with an 'environment', through actions that lead to changes in this environment and rewards that quantify the effect of these actions. The aim here is therefore to create strategies or 'policies' that optimise the total reward obtained.

Within this field there are numerous challenges, among which the "Restless Multiarmed Bandit Problem" stands out. In it, multiple agents or 'bandits' are considered, which can have two possible actions: be 'active' or 'passive'. Only a limited number of these bandits can be active at the same time, and depending on the action performed the change of state and the reward of all these processes can be different. It is therefore a problem of prioritization between the different stochastic processes, not only for the next time step, but also for the long-term future. This problem of 'resource allocation' has multiple practical applications such as the management of workload on servers [1] [2], the detection of channels in communications [3], the management of health systems [4] or in the dynamics of pricing [5]. However, due to the nature of the problem, a large number of 'visits' to each action/state pair are necessary for each bandit, which makes classical selection methods require very long convergence times and are ineffective. In this work, we employ a different policy, in which we index each state, thus setting a priority in activating the bandits: the Whittle index policy. This policy was proposed in 1988 by P. Whittle [6]. In our approach we proposed a new technique: the calculation of the indexes through two time scales to obtain convergence conditions under which our algorithm converges to an optimal policy.

In this paper we will start by introducing the basic concepts of Machine Learning such as the different types of learning such as 'supervised learning', 'supervised learning' and 'reinforced learning' and the differences between them, the Markov chains, the criteria of optimality and the 'value functions'. In the chapter 2, we will explain the origin of the 'index policies', their use in Reinforcement Learning and the Q-learning techniques used to model and learn from the problem environment. In the chapter 3, we will study two cases of Reinforcement Learning problems with their own dynamics: the circular dynamics problem and the restart problem. We will study the calculation of Whittle's indexes in an analytical way for these problems, the algorithm that we will use for its numerical calculation and in the chapter 3.2.1, we will make a scheme of the demonstration of the theoretical convergence of the algorithm. Finally, we will analyse the results obtained with this algorithm.

Índice general

Índice de figuras

Capítulo 1

Algoritmos de Machine Learning

1.1. Introducción histórica

Machine Learning es uno de los principales campos de Inteligencia Artificial (AI). El principal objetivo en este campo es la creación de modelos capaces de comprender una estructura de datos. Gracias a la flexibilidad de aplicaciones que tiene, es usado en multitud de áreas, como en la predicción de sistemas tales como préstamos bancarios en los que se calcula la probabilidad de un fallo en el pago, reconocimiento de imágenes o voz, diagnosis médicas, etc.

A pesar de ser un campo de *computer science*, difiere de los enfoques computacionales tradicionales. En la programación tradicional, los algoritmos son un conjunto de instrucciones explícitamente programadas y empleadas para el cálculo y solución de problemas. En Machine Learning, sin embargo, el algoritmo se entrena a partir de un set de datos y emplea un análisis estadístico para obtener estos valores. He aquí donde reside la fuerza de este rama de computer science: el empleo de instrucciones relativamente sencillas para automatizar cálculos muy complejos, con modelos no lineales que serían demasiado complicados de detallar expresamente.

Los orígenes de este campo se remontan a 1958, cuando Frank Roseblatt diseño la primera red neuronal artificial [7], llamada "Perceptrón", cuyo cometido original era el reconocimiento de patrones y formas. El año siguiente, Bernard Widrow y Marcian Hoff crearon un nuevo modelo de red neuronal llamado ADALINE [8] capaz de detectar patrones binarios, y por tanto, en una cadena de bits, predecir cual sería el valor del siguiente. El siguiente modelo, MADALINE [9], era capaz incluso de eliminar el eco en las llamadas telefónicas, siendo este la primera aplicación útil de las redes neuronales. En los años 60, R. J. Solomonoff introdujo los métodos Bayesianos [10] para la inferencia probabilística, hoy en día fundamental para la teoría básica de Machine Learning. Sin embargo, debido a la carga computacional necesaria para estas técnicas, se produjo un periodo de inactividad en la investigación de Machine Learning hasta 1982, cuando John Hopfield sugirió la creación de redes neuronales bidireccionales [11], similar a como funcionan en la realidad las neuronas. Sin embargo, no fue hasta 1990 y el siglo 21 en el que se empezaron a desarrollarse extensivamente, con el origen de los Support Vector Machines[12] y la popularización de las Recurrent Neural Networks (RNNs), en el que este campo realmente floreció. Actualmente, el desarrollo de Deep Learning[13] [14] ha permitido la aplicación de este campo en multitud de áreas.

1.2. Supervised vs Unsupervised vs Reinforcement Learning

Machine Learning puede clasificarse en tres tipos de algoritmos distintos: Supervised Learning, Unsupervised Learning y Reinforcement Learning, cada uno de ellos con funcionamientos y aplicaciones distintos.

En el primero de ellos, **Supervised Learning**, empleamos un set de datos de entrenamiento *etiquetados*, es decir, en el que cada muestra viene acompañada con una "variable objetivo". El objetivo es,

por tanto, diseñar un modelo capaz de analizar los datos entrantes y predecir que etiquetas tendrán a partir de los ejemplos dados en el entrenamiento. Estas variables pueden ser categóricas, en cuyo caso hablamos de un problema de **clasificación** como puede ser la clasificación de un email como 'Spam' o 'No Spam', o continuas, con un problema de **regresión**, como puede ser la predicción de los valores en Bolsa [15]. Los tres tipos de algoritmos más populares en este área son:

- **Neural Network**: Se trata de estructuras formadas por "neuronas". Cada una de ellas toma una serie de variables de entrada, realiza una combinación lineal sobre estas y es pasada por una *función de activación*, como puede ser la función sigmoide. En la actualidad, se diseñan redes con millones de neuronas [16] [17] distribuidas en "capas", donde el *input* de cada capa son los *output* de cada una de las neuronas de la capa siguiente.

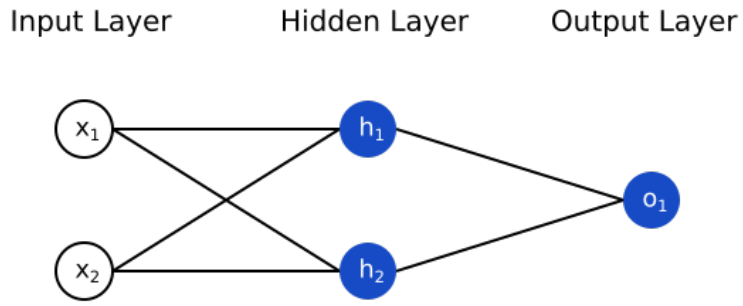
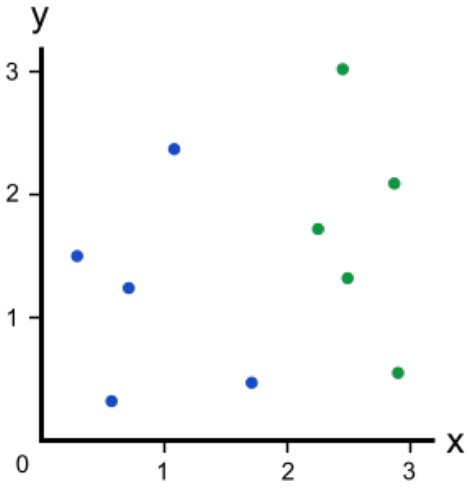
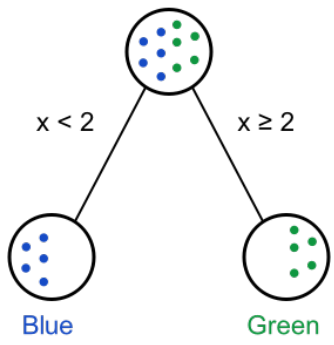


Figura 1.1: Ejemplo de un esquema de una red neuronal, formada por tres capas distintas: una primera capa, con las variables de entrada de los datos iniciales, una capa final con el valor a predecir y una capa intermedia, para aumentar la complejidad y potencia del modelo.

- **Random Forest**: Random Forest es una agrupación de árboles de decisión en el que en cada nodo se aplican unos criterios de clasificación a los datos. A través de sucesivos nodos, distribuidos en diferentes árboles de decisión con distintos criterios de clasificación, podemos parametrizar los datos de entrada en gran detalle.



(a) Ejemplo de distribución de datos clasificados por color.



(b) Nodo en un árbol de decisiones. En el, se obtiene las dos categorías de clasificación (azul y verde) en función de los valores de la variable x.

Figura 1.2: Ejemplo del funcionamiento de un árbol de decisiones empleado en Random Forest.

- **Support Vector Machines**: En un modelo de Support Vector Machines se emplea una representación en el espacio de variables de los datos del algoritmo, donde estos datos se encuentran clasificados en 2 o más categorías. El objetivo de este tipo de algoritmos es por tanto la creación de un hiperplano en este espacio capaz de separar estas categorías de modo que el margen entre ellas sea lo más ancho posible.

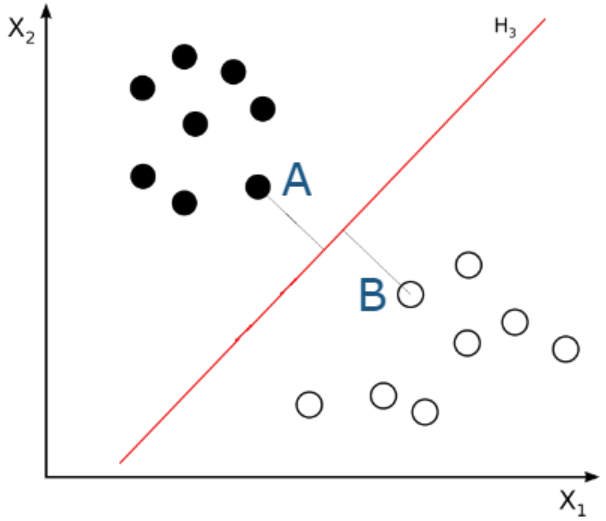


Figura 1.3: Clasificación de los datos a través de SVM. Cada una de las categorías, A y B, se encuentran separadas por un plano (línea roja) que maximiza el margen entre ambas clases.

Por otro lado, en **Unsupervised Learning** los datos no se encuentran etiquetados, de modo que el algoritmo tiene que encontrar semejanzas entre los datos. Los objetivos de Unsupervised Learning van desde el reconocimiento de patrones ocultos como el aprendizaje de características, en el que el algoritmo descubre nuevas representaciones en las que clasificar los datos. Una aplicación tradicional de este campo es en las transacciones. A partir de un set de datos con las compras que ha hecho un conjunto de clientes, podemos crear perfiles en los que clasificarlos en función del tipo de compras que realicen. Existen dos tipos fundamentales de algoritmos de Unsupervised Learning:

- **Clustering**: se denominan problemas de "clustering" aquellos en los que queremos descubrir agrupaciones inherentes en los datos. Las dos técnicas más comunes para esto son K-means, en la que agrupamos los datos en centroides intentando maximizar el número de datos en cada clúster.
- **Reglas de asociación**: El objetivo de este tipo de problemas es descubrir reglas que describan el comportamiento de un determinado conjunto de datos, tales como "la gente que compra X también tiende a comprar Y".

Reinforcement Learning se diferencia de estos otros dos paradigmas en la capacidad del algoritmo de aprender a través de las **reacciones de un entorno**. Por lo tanto, debemos distinguir dos elementos en cualquier problema de Reinforcement Learning: **el agente**, el cual realiza una acción en función del estado en el que se encuentre, y **el entorno**, que engloba a este agente y lo provee de nuevos estados y recompensas en función de las acciones que realice en el estado en que se encuentre este agente. Además de estos dos elementos, los algoritmos de Reinforcement Learning están formados por:

- **Política**: Define el comportamiento del agente en un momento dado. Se trata de un mapeo de cada estado percibido en el entorno con una de las acciones disponibles en ese estado.
- **Función de recompensa**: Cada acción realizada en cada estado por el agente esta recompensada por un valor específico. El objetivo en Reinforcement Learning es maximizar, no solo la

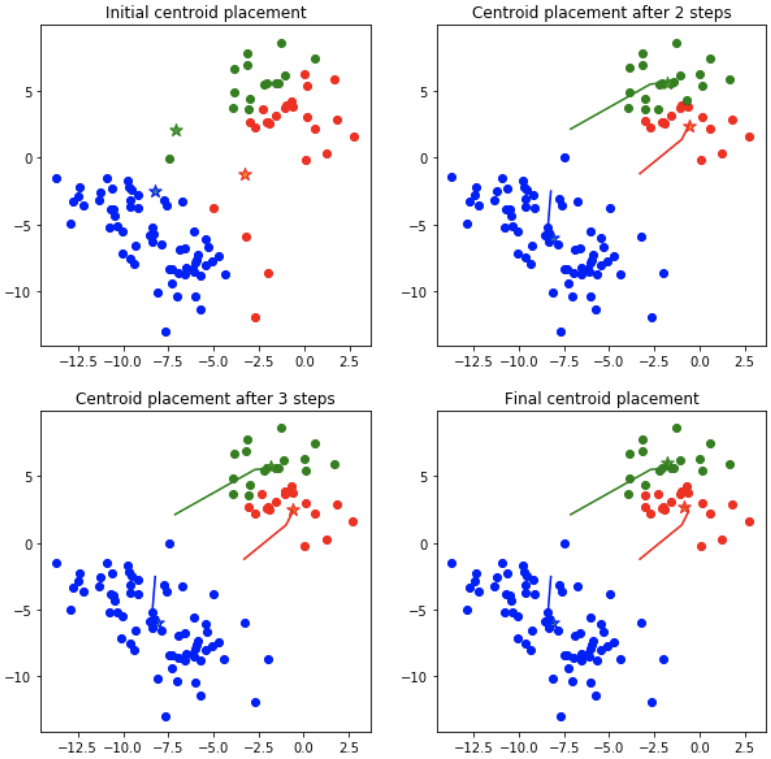


Figura 1.4: Ejemplo de clasificación a través de K-means: se establecen tres centroides aleatoriamente (los puntos estrellados) y en cada iteración, se desplazan estos centroides, asignando a los puntos más cercanos las categorías de estos centroides.

recompensa inmediata obtenida al realizar una acción, sino todas las recompensas futuras. Por lo tanto, la función de recompensa en un estado concreto cuantifica lo "bueno" que sea realizar una determinada acción en ese estado, mientras que la recompensa total define la calidad de la política.

- **Función de valor**: Si la recompensa nos indica lo buena que es una acción en un momento dado, la función de valor nos indica su efecto a largo plazo. Define, por lo tanto, la calidad de nuestra política. Definiremos en más detalle este parámetro en la sección 1.3.6

1.3. Introducción a las cadenas de Markov

Antes de introducir las cadenas de Markov, empezaremos con una introducción de algunas nociones importantes de teoría de probabilidad. Una variable aleatoria X es aquella cuyo valor está definido como el resultado de un fenómeno aleatorio. Un ejemplo de esto podría ser el resultado de tirar un dado o una moneda.

Definimos como un proceso aleatorio, también llamado "proceso estocástico", al conjunto de variables aleatorias indexadas por T, donde T puede ser una variable discreta (como el conjunto de los números naturales) o continua (el conjunto de números reales). Un ejemplo sería lanzar una moneda cada día, donde T es aquí el conjunto de días en el que se realiza el lanzamiento. El resultado de cada variable aleatoria dentro del proceso se puede considerar independientes de cada una, como el lanzamiento que hemos mencionado antes, o dependientes.

1.3.1. Cadenas de Markov

Existen numerosas familias de procesos estocásticos, como los procesos gaussianos, de Poisson, cadenas de Markov, etc. Todas las cadenas de Markov comparten una propiedad, la "Propiedad de

Markov", según la cual la distribución de probabilidad del valor futuro de una variable aleatoria depende únicamente de su valor en el presente, independientemente de sus valores en el pasado.

$$P(\text{future}|\text{present, past}) = P(\text{future}|\text{present})$$

Por lo que, sea una cadena de Markov definida como $X = (X_n)_{n\in\mathbb{N}} = (X_0, X_1, X_2, \dots)$, donde en cada instante de tiempo el proceso toma un valor discreto de un set de estados S tal que $X_n \in S, \forall n \in \mathbb{N}$, se cumple:

$$P(X_{n+1} = x_{n+1}|X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_2 = x_2, X_1 = x_1) = P(X_{n+1} = x_{n+1}|X_n = x_n)$$

Obsérvese en primer lugar que la caracterización completa de un proceso aleatorio de tiempo discreto que no verifique la propiedad de Markov puede ser engorrosa: la distribución de probabilidad en un momento dado puede depender de uno o varios instantes de tiempo en el pasado y/o el futuro. Todas estas posibles dependencias temporales hacen que cualquier descripción adecuada del proceso sea potencialmente difícil.

Sin embargo, gracias a la propiedad Markov, la dinámica de una cadena de Markov es bastante fácil de definir. De hecho, sólo necesitamos especificar dos cosas: una distribución de probabilidad inicial (es decir, una distribución de probabilidad para el instante de tiempo $T = 0$) denotada

$$\mathbb{P}(X_0 = s) = q_0(s) \qquad \forall s \in E$$

y una función de probabilidad de transición, que da las probabilidades de que un estado, en el momento n+1, suceda a otro, en el momento n, para cualquier par de estados, definida como

$$\mathbb{P}(X_{n+1} = s_{n+1}|X_n = s_n) = p(s_n, s_{n+1}) \qquad \forall (s_{n+1}, s_n) \in ExE$$

1.3.2. Matrices de probabilidad de transición

A la hora de realizar un cambio de estado, el entorno pasa al agente de un estado $S(t)$ a $S(t+1)$ con una determinada probabilidad, definida como:

$$P_{ss'} = \mathbb{P}\left[S_{t+1} = s'|S_t = s\right] \tag{1.1}$$

Si consideramos todos los posibles pares ss' del espacio de estados en el que se desarrolla el problema, podemos definir una matriz de probabilidad de transición de estados:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} \end{bmatrix} \tag{1.2}$$

Donde cada fila de la matriz representa la probabilidad de pasar desde el estado inicial al siguiente. Por lo tanto, la suma de todos los elementos de cada fila es igual a 1.

Las matrices de probabilidad de transición nos permiten calcular las transiciones de un estado a otro a lo largo de una cadena completa, gracias a la propiedad de Markov: Imaginemos que queremos calcular la probabilidad de llegar al estado 4 en $T = 4$ cuando en los tiempos anteriores $T = \{1, 2, 3\}$ hemos estado en los estados $1, 2, 3$ respectivamente. Gracias a la propiedad de Markov podemos desarrollar esta probabilidad como:

$$\begin{gathered} P_4 = \mathbb{P}\left[S_4 = 4|S_3 = 3, S_2 = 2, S_1 = 1\right] = \\ = \mathbb{P}\left[S_4 = 4|S_3 = 3\right] \cdot \mathbb{P}\left[S_3 = 3|S_2 = 2\right] \cdot \mathbb{P}\left[S_2 = 2|S_1 = 1\right] = \\ = p_{34} \cdot p_{23} \cdot p_{12} \end{gathered}$$

Donde en cada momento T, la probabilidad de pasar a un nuevo estado s' solo depende del estado presente s y no de los anteriores.

1.3.3. Procesos de recompensa de Markov: MRP

En la sección anterior hemos visto como los sistemas Markovianos están regidos por un conjunto de estados y una matriz de probabilidad de transición entre estos estados. Sin embargo, como hemos explicado en la sección 1.2, la recompensa obtenida en cada estado es fundamental para la obtención de políticas óptimas en Reinforcement Learning. Para ello, introduciremos primero los **Procesos de Recompensa de Markov** (MRP): cadenas de Markov con valores de juicio. En estas cadenas, obtenemos el valor de la recompensa de cada estado por el que pasa nuestro agente. Estas cadenas están definidas como:

$$R_S = \mathbb{E}[R_{t+1}|S_t] \tag{1.3}$$

En este proceso Markoviano estamos calculando la recompensa inmediata R_S que obtenemos para un determinado estado S_t. Estos procesos de recompensa solo tienen en cuenta la recompensa que se obtendrá en el siguiente paso. Sin embargo, en muchas ocasiones una política óptima implica unas primeras acciones que, si bien a corto plazo pueden no dar las mejores recompensas, permiten acceder a otros estados con recompensas que a largo plazo si que convierten a esa política en la mejor.

En función del tamaño de la cadena de Markov, se pueden considerar dos tipos distintos de "tareas":

- **Tareas episódicas**: Son problemas con un estado inicial y final bien definidos y que, por tanto, tienen un número de estados finito antes de que termine el proceso. Un ejemplo de este tipo de problemas es el de un coche autónomo, cuyo funcionamiento se limita a cada uno de los viajes que realiza y, por tanto, su estado inicial es el inicio de este viaje y el final es su llegada a su destinación. Una vez termina este proceso, se *reinicia* el problema, empezando con un nuevo estado inicial s_0 y sin tener en cuenta las recompensas obtenidas en el proceso anterior.

- **Tareas continuas**: No existe una condición definida bajo la cual termine el proceso estocástico y por lo tanto, si bien existe un estado inicial s_0, no existe un estado final. El número de estados puede ser, por lo tanto infinito. Un ejemplo de este tipo de problemas sería un termostato automático, capaz de regular la temperatura de la habitación con la finalidad de evitar que se tenga que regular manualmente. No existe una condición bajo la cual este proceso termine y por lo tanto puede seguir de forma indefinida un número indeterminado de estados.

En las tareas episódicas es sencillo considerar la recompensa total de cada estado de la cadena hasta llegar al estado final, ya que ésta es finita. Sin embargo, en las tareas continuas las cadenas no son finitas. Esto plantea varios problemas: ¿Cómo calculamos el retorno, la suma total de recompensas, como una cantidad finita en un proceso infinito? ¿Qué peso debería tener una recompensa que se pueda obtener en un futuro muy lejano en la toma de una acción en el presente? En la sección 1.3.4 estudiaremos métodos para responder a estas preguntas.

1.3.4. Criterios de optimalidad

Antes de introducir formalmente las funciones de valor, es necesario discutir el concepto de *optimalidad* en Reinforcement Learning. Tal y como introdujimos en la sección 1.2, una política define la actuación de un agente y, formalmente, es el mapeo de cada estado a la probabilidad de tomar cada acción. Si un agente sigue una política π en un tiempo t, entonces $\pi(a|s)$ es la probabilidad de que ese agente realice la acción a en el estado s en ese instante. El objetivo en Reinforcement Learning es la búsqueda de políticas que maximicen la recompensa obtenida por el agente *a largo plazo*, cantidad conocida como *retorno*. Este retorno se define, por tanto, como la suma de las recompensas que obtiene un agente al seguir una política π desde un estado inicial s_0. En [18] proponen tres criterios de optimalidad:

el criterio de *horizonte finito*, *horizonte infinito descontado* y *recompensa promedio*.

El criterio de **horizonte finito** (1.4) consideramos el valor esperado de la suma de todas las recompensas de una cadena. Todas estas recompensas están evaluadas con el mismo peso, de modo que una recompensa r tiene el mismo peso en el momento t que en un tiempo futuro $t+i$. Este tipo de criterios es empleado en las tareas episódicas, con un número determinado de transiciones T, donde la suma de un conjunto de recompensas finitas produce una recompensa finita. Sin embargo, este tipo de optimalidad no se puede aplicar en las cadenas continuas, ya que, aunque todas las recompensas sean finitas, la suma de estas es infinita.

$$E\left[\sum_{t=0}^{T} r_t\right] \tag{1.4}$$

El criterio de **horizonte infinito descontado** (1.5) considera el efecto de todas las recompensas de la cadena, incluso de las cadenas infinitas de las tareas continuas, empleando un término de descuento γ tal que $0 \leq \gamma < 1$. Este término se denomina *factor de descuento* y modela el hecho de que nuestro proceso estocástico no está seguro de si en la siguiente decisión el proceso puede o no terminar. Un ejemplo de esto sería que nuestro problema de "decision making" fuese un robot, y por lo tanto, el factor de descuento representa aquí la probabilidad de que el robot se desconecte en el instante siguiente. Este término a su vez regula el peso del valor de las recompensas a largo plazo: una misma recompensa, obtenida t iteraciones más tarde, tendrá un valor γ^t más pequeño que si sucede en el presente.

$$E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{1.5}$$

Esta expresión nos permite calcular el retorno total de una cadena de Markov incluso cuando esta es infinita. La demostración de esta convergencia es sencilla: Consideremos un $R_{\text{máx}} \geq r_t \quad \forall s(t) \in S$ tal que $R_{\text{máx}} < \infty$:

$$\sum_{t=1}^{\infty} \gamma^t r_t \leq \sum_{t=1}^{\infty} \gamma^t R_{\text{máx}} = \frac{\gamma}{1-\gamma} R_{\text{máx}} < \infty$$

Lo cual se cumple siempre que $0 \leq \gamma < 1$. En las tareas episódicas, la condición $\gamma < 1$ no es necesaria para la convergencia, aunque emplear $\gamma = 1$ sería equivalente a utilizar el criterio de horizonte finito en este caso. Si empleamos $\gamma = 0$, nuestro agente es *miope*, es decir, solo considera las recompensas inmediatas e ignora las recompensas futuras que pueda conseguir.

Este tipo de criterio de optimalidad es uno de los más empleados [18] y es el que emplearemos en este trabajo.

El último tipo de criterio de optimalidad que veremos es la **recompensa promedio** (1.6). Este tipo de criterio maximiza la recompensa promediada a largo plazo. En el caso anterior, cuando γ tiende a 1, el resultado converge al de este tipo de criterio. El principal problema que conlleva este criterio es que para cadenas infinitas, no podemos distinguir entre dos políticas en las que una reciba muchas recompensas en las fases iniciales y en la otra no. Esta diferencia inicial se encuentra oculta por el promediado.

$$\lim_{T\to\infty} E\left[\frac{1}{T}\sum_{t=0}^{T} r_t\right] \tag{1.6}$$

Como hemos visto previamente, la decisión de que tipo de criterio elegir depende especialmente del tipo de problema a resolver: si se trata de una tarea episódica, el modelo de horizonte finito es el más conveniente, mientras que para una tarea continua, el modelo de horizonte infinito descontado es mejor, gracias a que asegura la existencia de, al menos, una política óptima estcionaria y determinsta, mientras

que en los otros casos las políticas óptimas generalmente dependen del instante temporal, de modo que no son estacionarias [19].

1.3.5. Procesos de decisión de Markov

En las secciones anteriores hemos introducido el papel de las recompensas en las cadenas de Markov y como calcular el retorno total incluso en cadenas infinitas. Sin embargo, para poder definir una política dada, es necesario que el agente sea capaz de decidir una *acción* en cada nuevo estado en el que se encuentre. Estas acciones entran en el marco de los procesos de Markov en la forma de **Procesos de Decisión de Markov** o MDP: procesos de recompensas de Markov como los descritos en la sección 1.3.3 en los que se realizan acciones. Estas acciones pueden afectar la recompensa y al estado al que avanza el agente. Bajo este paradigma, las matrices de transición y las funciones de recompensa reciben una nueva dependencia con la variable de *acción*:

$$P^a_{ss'} = \mathbb{E}\left[S_{t+1} = s' | S_t = s, A_t = a\right]$$
$$R^a_S = \mathbb{E}\left[R_{t+1} = R | S_t = s, A_t = a\right]$$

Bajo estas nuevas dependencias, la matriz de transición y la recompensa puede variar en función de la acción a que se tome. Por otro lado, siempre podemos recuperar un proceso de Markov o una MRP a partir de un proceso de decisión de Markov: Sea un MDP formado por la tupla (S,A,P,R) y una política π, la secuencia de estados $S_1, S_2, \ldots$ es un proceso de Markov (S,P) bajo una política determinada π. De la misma forma, la secuencia $(S_1,R_1),(S_2,R_2),\ldots$ es una MRP formada por la tupla (S,P,R) cuya matriz de transición de estados es

$$P^\pi_{ss'} = \sum_{a \in A} \pi(a|s) P^a_{ss'}$$

Las distintas acciones que se realiza en MDP en función de los estados definen las políticas π de estos procesos. En la siguiente sección, discutiremos como evaluar estas políticas y definiremos la *política óptima*: la política con el mayor retorno posible.

1.3.6. Funciones de valor y ecuación de Bellman

En la sección 1.2 introdujimos el concepto de "función de valor": se trata de una estimación de la "bondad" de un agente dado en función de su política a largo plazo, en función del conjunto de recompensas que consigue al realizar varias acciones en varios estados. Sin embargo, aquí debemos realizar la distinción de dos tipos de funciones de valor: la función V, que estima la bondad de *estar* en un estado, y la función Q, referida comúnmente como *Q-value*, que estima la bondad de *realizar una acción en un estado*. De esta manera, la función V será solo función del estado en el que nos encontremos, $V(s)$, mientras que la función Q dependerá tanto del estado como de la acción, $Q(s,a)$.

El *valor de un estado s bajo una política* π, denotada $V^\pi(s)$, es el retorno esperado, partiendo de un estado s y siguiendo una política π. Empleando el criterio del horizonte infinito descontado (1.5), podemos expresar esta función como:

$$V^\pi(s) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right\} \tag{1.7}$$

Por otro lado, *la función de valor acción-estado* está definida como el retorno obtenido al empezar desde un estado s, efectuando una acción a y continuando con una política π:

$$Q^\pi(s,a) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a \right\} \tag{1.8}$$

La principal característica de este tipo de funciones es su capacidad de escribirse de forma recursiva [20]:

$$\begin{aligned} V^{\pi}(s) &= E_{\pi}\left\{r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s\right\} \\ &= E_{\pi}\left\{r_t + \gamma V^{\pi}(s_{t+1}) | s_t = s\right\} \\ &= \sum_a \pi(a|s) \sum_{s'} P^{\pi}(s,s')\left[r + \gamma V^{\pi}(s')\right] \end{aligned} \tag{1.9}$$

Donde $P^{\pi}(s,s')$ es la probabilidad de transición de un estado s a un estado s' bajo la política π, tal y como hemos discutido en la sección 1.3.5 y $R(s,a,s')$ es la recompensa obtenida al realizar la acción a para pasar de un estado s a un estado s'.

El objetivo para cualquier cadena de Markov en Reinforcement Learning es realizar la mejor política π, es decir, aquella que maximice la recompensa bajo un criterio de optimalidad dado. Se define *política óptima* π^* a aquella cuya función de valor $V^{\pi^*}(s) \geq V^{\pi}(s)$ para todos los estado s del espacio de estados S y para todas las políticas π. La expresión óptima de la ecuación (1.9) satisface:

$$V^{\pi^*} = \max_{a \in A} \sum_{s' \in S} P^a(s,s')\left(R(s,a,s') + \gamma V^{\pi^*}(s')\right) \tag{1.10}$$

Esta ecuación, llamada *Ecuación de optimalidad de Bellman*, expresa la relación entre el valor de un estado y los valores de los estados siguientes y establece que el valor de un estado bajo una política óptima debe ser igual al retorno esperado para la mejor acción en ese estado. Por lo tanto, una acción óptima, dada una función de valor óptima $V^* = V^{\pi^*}$ es:

$$\pi^*(s) = \arg\max_a \sum_{s' \in S} P^a(s,s')\left(R(s,a,s') + \gamma V^*(s')\right) \tag{1.11}$$

Mientras que una acción *greedy* elige solamente aquella acción que maximiza la recompensa a un solo paso, es decir, que maximiza $P(s,s')R(s,a,s')$ en una única transición $s \to s'$, la política óptima π^* maximiza el conjunto de todas las futuras decisiones.

De forma análoga, para la función acción-estado Q, tenemos:

$$Q^*(s,a) = \sum_{s'} P^a(s,s')\left(R(s,a,s') + \gamma \max_{a'} Q^*(s',a')\right) \tag{1.12}$$

A partir de las definiciones de las funciones $Q(s,a)$ y $V(s)$, podemos ver una relación entre ambas: La función de valor $Q(s,a)$ establece de forma explícita el parámetro de acción a en la función $V(s)$. Es por ello que, siguiendo una política π, la relación entre Q^{π} y V^{π} es:

$$V^{\pi}(s) = \sum_{a \in A} \pi(a|s) \cdot Q^{\pi}(s,a) \tag{1.13}$$

Es decir, sumamos los valores de los pares acción-estado por la probabilidad de tomar una acción a en un determinado estado s, a partir de la política π.

Los diagramas de la imagen 1.5 muestran los tramos de los futuros estados y acciones consideradas para las ecuaciones de Bellman (1.10) (a la izquierda) y (1.12) (a la derecha). Cada circulo blanco representa un estado y cada círculo negro representa un par acción-estado. En el diagrama de la izquierda, partiendo del estado s, el agente evalúa cada una de las acciones y toma la óptima a partir de su política π. A partir de esta acción, el entorno puede responder con varios nuevos estados s' con una recompensa r definida para cada estado, en función de la dinámica del problema. Por otro lado, en el diagrama de la

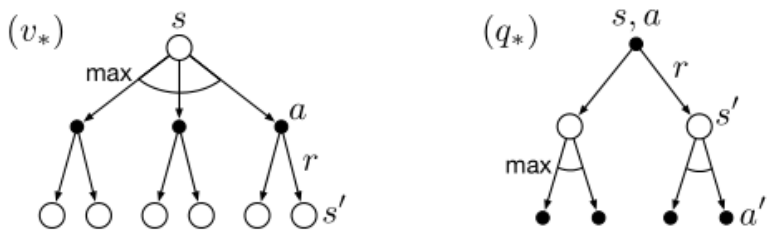


Figura 1.5: Diagramas para las funciones V^* y Q^*.

derecha, a partir del par acción-estado, pasa a un nuevo estado s' con recompensa r, desde el cual elige, a partir de su política π, la acción óptima a'.

Para las cadenas de Markov finitas, la ecuación de Bellman para V^* (1.10) tiene una solución única. La ecuación de optimización de Bellman es en realidad un sistema de ecuaciones, una para cada estado, de modo que si hay n estados, entonces existen n ecuaciones con n incógnitas. Si se conoce la dinámica del entorno, se puede resolver este sistema de ecuaciones para V^* utilizando cualquiera de los diversos métodos para resolver sistemas de ecuaciones no lineales. Una vez calculado V^*, es sencillo determinar la política óptima: para cada estado s siempre habrá una o mas acciones a que maximicen la ecuación de optimalidad de Bellman, de modo que cualquier política que asigne una probabilidad no nula solo a estas acciones es una política óptima. De esta forma, partiendo de la función de valor óptima V^*, la mejor acción en el siguiente paso será la acción óptima. Este tipo de políticas se denominan *greedy*, ya que seleccionan acciones basadas en consideraciones inmediatas, sin tener en cuenta la posibilidad de que esa acción no permita acceder a recompensas mejores en el futuro.

Por otro lado, elegir acciones óptimas con Q^* es mucho más eficaz: en este caso, el agente no realiza una búsqueda en el paso temporal siguiente, como ocurre con V^*, sino que para cada estado s, simplemente busca la acción que maximice $Q^*(s,a)$. Esta función de valor no solo nos da el resultado óptimo para el siguiente paso temporal, sino para todas las futuras transiciones, ya que provee con el retorno óptimo esperado en cada par acción-estado.

Capítulo 2

Política de índices en Reinforcement Learning

2.1. Introducción al índice de Gittins e índice de Whittle

Dentro de las familias de problemas de Decisión Markovianos, existen dos de especial interés en Reinforcement Learning: el "Multi-Armed Bandit Problem" (MABP) y "Restless Multi-Armed Bandit Problem" (RMABP). En estos problemas el Bandido es un término genérico para referirse a un solo proceso de Markov, con un espacio de estados S y de acciones A. De esta forma, este tipo de problemas plantea un nuevo paradigma con respecto al introducido en el capítulo 1: la gestión de múltiples procesos estocásticos, o bandidos, simultáneamente.

En el "Multi-Armed Bandit Problem", consideramos N procesos de Markov simultáneos en los que se pueden considerar dos posibles acciones: *activar* el proceso ($a = 1$) o mantenerlo *pasivo* ($a = 0$). En este caso, de los N procesos, solo uno puede establecerse como activo en cada momento mientras que el resto deben mantenerse pasivos. A su vez, solamente el proceso activo puede cambiar de estado y obtener una recompensa por ello, mientras que los estados pasivos se mantienen "congelados", sin cambios de estado ni recompensas. Este tipo de problema se puede considerar como un problema de "asignación". Uno de los primeros trabajos en intentar plantear una solución para este problema surgió en 1960 [21] con el objetivo de diseñar modelos matemáticos que definieran "políticas de parada", es decir, sistemas que se mantuviesen "activos" hasta que estarlo dejara de ser rentable. Sin embargo, no fue hasta los años 70 en el que Gittins y sus colaboradores obtuviesen la solución óptima para este problema [22]. En ella, plantean una nueva variable, el índice de Gittins $\lambda(s) \in \mathbb{R}, \forall s \in S$, tal que la política óptima para el problema de asignación es elegir el bandido i tal que

$$i_t = \underset{i \in \{1,\dots,n\}}{\arg\max} \{\lambda_i(s_i)\}$$

Es decir, en cada momento t, activar el bandido i tal que el índice λ de ese bandido en el estado s fuese el mayor con respecto al resto de bandidos. Gittins partió del concepto de *tiempo de parada*, similar al planteado en [21] para diseñar los índices:

$$\lambda_i(s_i) = \sup_{\tau > 0} \frac{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \gamma^t r_i(s_i(t)) | s_i(0) = s_i\right]}{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \gamma^t | s_i(0) = s_i\right]} \tag{2.1}$$

Donte τ es el tiempo de parada del proceso de Markov, es decir, el momento en el que este proceso pasaría de ser "activo" a "pasivo" y en el numerador tenemos la recompensa descontada hasta el momento τ, empleando el criterio de optimalidad de "horizonte infinito descontado" introducido en la sección 1.3.4 y en el denominador el tiempo descontado hasta el momento τ. Por lo tanto, $\lambda_i(s_i)$ es la **máxima recompensa por unidad de tiempo**, la "densidad de recompensa".

Por otro lado, en el segundo problema, el "Restless Multi-Armed Bandit Problem", plantea una generalización de los MABP: en este caso, consideramos que podemos activar en cada momento $K < N$ bandidos de modo que estos bandidos activos cambian de estado y obtienen recompensas acorde. La diferencia con respecto al primer caso, además del generalizar el número de bandidos que podemos activar, es el hecho de que aquellos bandidos que se encuentran "pasivos" también pueden cambiar de estado, con una dinámica distinta a la de los bandidos "activos" y con una función de recompensa distinta. Si bien el problema del Multi-Armed Bandit es un caso particular del Restless Multi-armed Bandit Problem, no fue hasta 1988 en el que Whittle [6] desarrolló una política de índices similar a la de Gittins capaz de gestionar este problema. Este cálculo requiere del conocimiento total de las matrices de transición del sistema y, por tanto, de la dinámica del problema. Para la mayoría de las aplicaciones prácticas, tales como la gestión de la programación de tareas en cloud computing [2], detección de canales en comunicaciones [3], sistemas de salud [4] o en la dinámica de la fijación de precios [5], esta situación no se suele dar por parte de los controladores de los sistemas.

Q-learning, desarrollado por Watkins en 1989[23], es uno de los métodos "model free" clásicos más empleados en el campo de Reinforcement Learning. Se trata de un algoritmo capaz de generar una política que pueda de gestionar cambios de estados y recompensas, en la que se controlan las acciones que toma un agente. Tal y como demuestran en [24], Q-learning es apto de alcanzar una política óptima maximizando la recompensa descontada en el horizonte infinito. Emplearemos este técnica para obtener un análisis del comportamiento del sistema con el que calcular los índices.

En la siguiente sección, analizaremos en detalle esta política y su implementación en nuestro trabajo.

2.2. Índices de Whittle para familias de cadenas de Markov

Sea una cadena de Markov en un espacio de estados finito $S = \{1,2,\ldots,d\}$, con un espacio de acciones definido como $A = \{0,1\}$ y unas probabilidades de transición $p(i,a,j)$ que representen la probabilidad de transición del estado i al estado j realizando la acción a, con $i,j \in S$ y $a \in A$. Dado que estas cadenas son *cerradas*, desde cualquier estado i la suma de las probabilidades de transición a todos los posibles estados j suman 1, satisfaciendo entonces $\sum_j p(i,a,j) = 1$. Este espacio de acciones es binario y tiene la interpretación de *activar* o dejar *pasivo* ese proceso estocástico.

Consideremos ahora N procesos estocásticos, todos ellos con el mismo espacio de estados S y con las matrices de transición $P^1, P^0 \in [0,1]^{|S|\times|S|}$ en función de si tomamos una acción "activa" ($a = 1$) o "pasiva" ($a = 0$). Dado que tenemos N procesos estocásticos distintos sucediendo al mismo tiempo, definiremos $\mathbf{S}(t) = (s_n(t) : n \in [N])$ y $\mathbf{A}(t) = (a_n(t) : n \in [N])$ como el vector de estados y acciones que realizan los N bandidos en un momento determinado t.

En cada instante t, activamos siempre $K \leq N$ proyectos de modo que

$$\sum_{n\in[N]} \mathbf{A}(t) = K \tag{2.2}$$

Es decir, dado que $a_n(\cdot) = 1$ indica que el bandido n está activo, mientras que los bandidos pasivos están representados como $a_n(\cdot) = 0$, se cumple que para cada instante t, la suma de todos los valores del vector $\mathbf{A}(t)$ es igual al número de bandidos activos K.

Sin embargo, la condición (2.2) es demasiado restrictiva como para que pueda resolverse este problema fácilmente. La propuesta de Whittle [6] para resolver esto fue una versión relajada de esta condición, sustituyendo la necesidad de que se cumpla 'en cada instante de tiempo' a solo 'en promedio':

$$\liminf_{n\uparrow\infty} \frac{1}{n}\mathbb{E}\left[\sum_{m=0}^{n-1} \mathbf{A}_m\right] = M \tag{2.3}$$

Bajo esta condición más relajada, el problema puede resolverse siempre y cuando cumpla la **condición de indexabilidad** [6] [25]. Para explicar esta condición, consideremos primero un proceso de Markov de un solo bandido, con unas probabilidades de transmisión $P_{a,b}$. En cada instante de tiempo,

se pueden realizar dos acciones posibles $a \in \{0(\text{pasivo}), 1(\text{activo})\}$. Bajo este sistema, consideraremos un *subsidio por pasividad*, donde las recompensas de las acciones pasivas cuentan con un añadido λ, es decir:

$$R(s) = \begin{cases} R_1(x) & a = 1 \text{ activo} \\ R_0(x) + \lambda & a = 0 \text{ pasivo} \end{cases}$$

Bajo este nuevo sistema, la función de valor de la ecuación (1.10) pasaría a ser:

$$V(s) = \text{máx}\left(R_1(s) + \gamma \cdot \sum_j p(k,1,j)V(j), \quad R_0(s) + \lambda + \gamma \cdot \sum_j p(k,0,j)V(j) \right) \tag{2.4}$$

$$= \max_{a \in \{0,1\}} \left[u\left(R_1(s) + \gamma \cdot \sum_j p(k,1,j)V(j) \right) + (1-u)\left(R_0(s) + \lambda + \gamma \cdot \sum_j p(k,0,j)V(j) \right) \right] \tag{2.5}$$

De esta forma, la acción activa es óptima cuando $R_1(s) + \gamma \cdot \sum_j p(k,1,j)V(j)$ es el máximo mientras que la acción pasiva será óptima si $R_0(s) + \lambda + \gamma \cdot \sum_j p(k,0,j)V(j)$ es máximo.

A partir de la relación entre las funciones de valor de estado $V(s)$ y de acción-estado $Q(s,a)$ definida en (1.13), podemos calcular los *Q-value* de la función de acción-estado $Q(s,a)$ como:

$$Q(s,a) = a\left(R_1(s) + \gamma \cdot \sum_j p(k,1,j)V(j) \right) + (1-a)\left(R_0(s) + \lambda + \gamma \cdot \sum_j p(k,0,j)V(j) \right) \tag{2.6}$$

Sea $\Pi(\lambda)$ el subconjunto de estados $s \in S$ en los que la acción pasiva es óptima bajo el subsidio λ, es decir

$$\Pi(\lambda) = \left\{ s \in S : R_0(s) + \lambda + \gamma \cdot \sum_j p(k,0,j)V(j) \geq R_1(s) + \gamma \cdot \sum_j p(k,1,j)V(j) \geq V(j) \right\}$$

Un bandido formado por la tupla (S, A, P, R, γ) es indexable si $\Pi(\lambda)$ es creciente en λ, es decir

$$\lambda_1 \geq \lambda_2 \Rightarrow \Pi(W_1) \supseteq \Pi(W_2) \tag{2.7}$$

Por lo tanto, un bandido es indexable si a medida que aumentamos el nivel de subsidio pasivo, también lo hace el número de estados para el cual esa acción es óptima.

Por otro lado, sea un bandido $(S, P^1, P^0, R^1, R^0, \gamma)$ indexable, su índice de Whittle $g : S \to \mathbb{R}$ está definido como:

$$g(s) = \text{ínf}\{\lambda : s \in \Pi(\lambda)\}, s \in S \tag{2.8}$$

Es decir, de todos los posibles subsidios λ que se podrían aplicar al estado s para que este pasara a formar parte del conjunto $\Pi(\lambda)$, el índice de Whittle $g(s)$ es el valor mínimo de λ capaz de hacer esto.

Por lo tanto, para un valor de λ determinado, la recompensa obtenida por activar o no ese bandido es exactamente la misma. Con valores de λ mayores, la política óptima sería dejarlo pasivo, ya que sería la acción con mayor recompensa según la ecuación (2.4), mientras que para valores menores, la acción óptima sería activar el bandido. Es por ello que el valor $g(s)$, que delimita la rentabilidad entre ambas acciones, actúa como un índice para cada estado, definiendo así su *prioridad* a la hora de activar ese bandido o no. Para el caso frontera entre ambas acciones, este valor $\lambda = g(s_i)$ actúa como la diferencia de rentabilidad entre ambas acciones, es decir:

$$g(s) = \left(R_1(s) + \gamma \cdot \sum_j p(k,1,j)V(j) \right) - \left(R_0(s) + \gamma \cdot \sum_j p(k,0,j)V(j) \right) = Q(s,1) - Q(s,0) \tag{2.9}$$

A través de la **heurística del índice de Whittle**, si un *restless bandit* es indexable con $g_i : S_i \to \mathbb{R}$ el índice de Whittle para el bandido $i, 1 \leq i \leq N$ opera de tal manera que en cada iteración aplicamos la acción activa a los K bandidos con el mayor índice $W_i(s_i(t))$ y la acción pasiva a los $N-K$ bandidos restantes.

Bajo la condición relajada (2.3), Whittle [6] demostró que esta heurística era óptima. Esta aproximación, sin embargo, en el problema original con (2.2) es cuasióptima, es decir, si bien es una política con un retorno muy alto, no es óptimo como en el caso de Q-learning. Sin embargo, la gran ventaja de esta heurística es la capacidad de reducir la complejidad del problema de ser exponencial con N a ser solo lineal con este [6]. Empleando la heurística original de Q-learning, a partir de un número relativamente pequeño de bandidos, incluso con un espacio de estados limitado, el problema se vuelve demasiado exigente computacionalmente. Es aquí donde radica la potencia de estas políticas de índices: la capacidad de subdividir el espacio de estados de un MDP que crece exponencialmente con el número de bandidos en múltiples cadenas individuales de Markov, donde el espacio de estados total crece linealmente con este numero de bandidos, evitando así la "maldición de la dimensionalidad" presente en el primer caso.

En nuestro trabajo, calcularemos los términos $Q(s,a)$ y $g(s)$ para obtener una política óptima. Como podemos ver en las ecuaciones (2.6) y (2.9), los términos $Q(s,a)$ y $g(s)$ se encuentran acoplados, de tal manera que para calcular uno necesitamos los valores del otro. Con el fin de desacoplar este sistema y poder calcular fácilmente estos índices, a la vez que introducir unas condiciones de convergencia para el cálculo numérico de éstos, emplearemos una aproximación adiabática, donde el cálculo del índice $g(s)$ se realiza en una escala más lenta que la de $Q(s,a)$. Este tipo de aproximaciones son comunes en física y química molecular [26] [27] [28], donde se pretende desacoplar sistemas de ecuaciones diferenciales, en estos casos referidos a la dinámica de los electrones y los núcleos de los átomos. De la misma forma que para los electrones el movimiento del núcleo de los átomos es tan lento que se considera estático [29], de cara a la variable $Q(s,a)$, el índice $g(s)$ será cuasi-estático.

2.3. Q-learning

2.3.1. Cálculo iterativo de los Q-values iterativo

Como hemos visto en la sección 1.3.6, la principal característica de la función de valor acción-estado, también llamado Q-value, es la capacidad de evaluar no solamente la "bondad" del siguiente paso temporal, como ocurre con la función $V(s)$ que sigue una política *greedy*, sino que es capaz, en cada estado, de evaluar los futuros estados siguiendo una misma política π. Para hacer esto, el algoritmo no necesita saber de antemano la dinámica del sistema y las matrices de transición, sino a partir de la interacción con el entorno. Este método fue introducido en 1989 por Watkins [23] a través de la siguiente fórmula:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t) \right] \tag{2.10}$$

En este caso, la función aprende el Q-value del par acción-estado s_t, a_t de forma iterativa, empleando un tamaño de paso $0 < \alpha < 1$. En este modelo de aprendizaje, no necesitamos conocer explícitamente la política π, sino solamente simular el cambio de estado al realizar una acción a desde el estado s. A través de este método, se ha demostrado [30] la convergencia del Q-value $Q(s,a)$ a su valor óptimo $Q^*(s,a)$.

En este esquema introducimos el concepto de *política de control ε-greedy*. Analizaremos este concepto en detalle en la sección 3.2. Por ahora, nos basta con aclarar que se trata de una política empleada para elegir que acciones realizar para cada estado S del entrenamiento que asegura una cierta diversidad, de modo que tengamos experiencia suficiente sobre todos los pares acción-estado.

Una de las principales condiciones de convergencia [31] que debe cumplir Q-learning está relacionado con el tamaño de los pasos α. Para garantizar la convergencia, $\{\alpha\}$ debe decrecer con respecto al número de pasos n como $\sum_n^\infty \alpha(n) = \infty$ y $\sum_n^\infty \alpha(n)^2 < \infty$. Detallaremos la demostración completa de la convergencia al valor óptimo en la sección 3.2.1

```
Parámetros del algoritmo: tamaño de paso 0 < α ≤ 1, ε pequeño
Inicializamos Q(s,a) para todos los s∈S y a∈A arbitrariamente

Bucle para cada episodio:
    Inicializamos S
    Bucle para cada paso en el episodio:
        Elegir acción A para el estado S a partir de una política de control
            (ε−greedy)
        Realizar acción A, obtener recompensa R y nuevo estado S'
        Q(S,A) ← Q(S,A) + α[R + γ máx_a Q(S',a) − Q(S,A)]
        S ← S'
    Repetir hasta que S sea el último estado del episodio
```

Figura 2.1: Esquema de aprendizaje por Q-learning

2.3.2. Q-learning con política de índices

En la sección anterior hemos visto el algoritmo general para el cálculo iterativo de los Q-values a través de la interacción con el entorno. A continuación, introduciremos las nociones de políticas de índices planteadas en la sección 2.2 para generar una nueva expresión combinando las ecuaciones (2.6) y (2.10).

En la ecuación (2.10) actualizamos el valor de cada Q-value para cada par acción-estado como una combinación lineal entre el valor original y un nuevo término $R_{t+1} + \gamma \max_a Q(s_{t+1}, a)$. En nuestro nuevo algoritmo, introducimos la posibilidad de obtener o no un subsidio extra en forma del índice de Whittle $g(s_t)$ en función de si la acción a que realizamos es pasiva o activa:

$$Q^x_{n+1}(s_t, a_t) \leftarrow Q^x_n(s_t, a_t) + \alpha(n) \left[(1 - a_t)(R_0(s_t) + g_n(x)) + a_t R_1(s_t) + \gamma \max_{v \in \{0,1\}} Q^x_n(s_{t+1}, v) - Q^x_n(s_t, a_t) \right] \tag{2.11}$$

En esta expresión, calculamos el Q-value del par acción/estado a_t, s_t en la iteración $n+1$ a partir de su valor en la iteración n, su recompensa, sea esta $R_1(s_t)$ si la acción es activa o $R_0(s_t) + g_n(x)$ si es pasiva y el mayor Q-value de entre las dos acciones posibles para el siguiente estado.

En la ecuación (2.11) introducimos el índice de Whittle $g_n(x)$ para el estado $x \in S$, donde ser activo o pasivo es igualmente deseable. En nuestro proceso de Q-learning utilizamos dos variables de estado: por un lado, s_t representa el estado siendo visitado por el agente que es el bandido en cada iteración, mientras que x es una "prospección" a todos los posibles estados del conjunto S. Para el valor actual de s_t, calculamos $Q^x_{n+1}(s_t, u_t)$ para todos los posibles valores de x.

En la ecuación (2.11) introducimos el uso del índice de Whittle $g_n(x)$. Este índice se debe actualizar a lo largo de las n iteraciones en las que se entrena nuestro algoritmo simulando nuevos estados y recompensas a través del proceso de Q-learning. Sin embargo, debido al acoplamiento entre los Q-value y estos índices, es difícil garantizar la convergencia actualizando ambos al mismo tiempo. Para ello, implementamos una segunda escala de tiempo en la cual calculamos estos índices. Esta escala de tiempo irá más lenta que la escala normal, en la que actualizamos los valores de los Q-values con (2.11), de tal manera que en el cálculo de los Q-value, los índices de Whittle se considerarán cuasi-estáticos ya que estos se actualizan mucho menos a menudo. Para esto, introducimos otra secuencia de tamaños de paso $\{\beta(n)\}$ que, al igual que con $\{\alpha(n)\}$, satisface $\sum_n^\infty \beta(n) = \infty$ y $\sum_n^\infty \beta(n)^2 < \infty$. Combinando las actualizaciones iterativas de (2.10) con la fórmula de los índices (2.9), obtenemos la siguiente expresión:

$$g_{n+1}(x) = g_n(x) + \beta(n) \left(Q^x_n(x, 1) - Q^x_n(x, 0) \right) \tag{2.12}$$

Los tamaños de paso $\alpha(n)$ y $\beta(n)$ tienen ambos que cumplir las condiciones de convergencia que hemos descrito antes, y al mismo tiempo permitir que el índice $g(x)$ se actualice en una escala de tiempo más lenta que la del cálculo de $Q^x(s_t, u_t)$. Trabajos previos [31] han dado como resultado el empleo de las siguientes secuencias de tamaños de paso, con características vitales para la convergencia del algoritmo que discutiremos en detalle en el capítulo 3.2.1:

$$\alpha(n) = \frac{1}{\lceil \frac{n}{500} \rceil} \tag{2.13a}$$

$$\beta(n) = \frac{1}{1 + \lceil \frac{n \log n}{500} \rceil} I\{n(\mathrm{mod} N) \equiv 0\} \tag{2.13b}$$

Donde $\beta(n) \neq 0$ solamente en aquellas iteraciones n que sean múltiplo del número de bandidos presentes en el problema, N. Por lo tanto, cuanto mayor sea el número de bandidos empleados, mayor diferencia habrá entre ambas escalas de tiempo.

Capítulo 3

Cálculo de los índices de Whittle

3.1. Valor teórico de los índices de Whittle

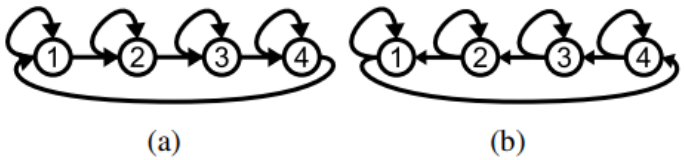
Antes de proceder a realizar el cálculo numérico de los índices de Whittle, primero calcularemos el valor teórico de estos índices para dos casos distintos. Primero, consideraremos uno de "dinámica circular", en el que los bandidos se mueven a través de una cadena de estados en la que al llegar a un extremo, pasan a estar en el extremo opuesto. El segundo problema que exploraremos es el "problema con reinicio", donde los bandidos pueden avanzar un estado o volver al primer estado de la cadena.

3.1.1. Dinámica circular

En el caso de la dinámica circular, consideraremos un espacio de estados $S = \{1,2,3,4\}$ donde las matrices de transición de estados para la acción activa ($u = 1$) y pasiva ($u = 0$) son:

$$P_0 = \begin{pmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix} \qquad P_1 = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \end{pmatrix}$$

Es decir, tomando una acción pasiva en un estado s, el bandido tiene 50 % de probabilidades de permanecer en ese estado y un 50 % de retroceder al estado $s-1$. Si ese estado es el estado 1, retroceder supondría pasar al estado 4. Análogamente, tomando una acción activa, el bandido tendría un 50 % de probabilidades de permanecer en ese estado y un 50 % de avanzar al estado siguiente, donde, en caso de estar en el estado 4, pasaría al estado 1.



(a) (b)

Figura 3.1: Visualización de la dinámica de los bandidos en el problema de la dinámica circular, para la acción activa (a) y pasiva (b)

La recompensa en este modelo no depende de la acción que tomen los bandidos, sino únicamente del estado en el que esté, y está definida como $R(1) = -1, R(2) = 0, R(3) = 0, R(4) = 1$. Estudios previos

[32], empleando el criterio de recompensa media en el horizonte infinito (ecuación (1.6)), demostraron que los índices de Whittle convergían a los valores $g(1) = -1/2, g(2) = 1/2, g(3) = 1$ y $g(4) = -1$: El algoritmo prioriza activar aquellos bandidos que se encuentren en el estado 3 por encima de cualquier otro, ya que en caso de avanzar a un estado, pasaría al 4 con recompensa $R(4) = 1$. Por otro lado, el último bandido que activaría sería aquel en el estado 4 ya que, de hacerlo, este podría pasar al estado 1 con recompensa $R(1) = -1$.

Estos valores exactos de los índices dependen del criterio de "optimalidad" empleado. En nuestro caso, con el criterio de recompensa descontada en el horizonte infinito de tiempo, el valor de estos índices dependerá del parámetro de descuento γ empleado. Sin embargo, en el límite $\gamma \to 1$, nuestros valores deberían converger a los aquí citados previamente. Lo que debería permanecer igual, en cualquier caso, es el orden de estos índices, es decir: primero activar el estado 3, luego 2, el 1 y por último el 4. Siguiendo con esa intuición, podemos calcular los índices de Whittle para el criterio de recompensa descontada enpleando las ecuaciones (2.4) y (2.9), activando secuencialmente los estados en el mismo orden que el citado previamente.

- **Estado 3**: Empezamos con todos los estados pasivos. En la ecuación (2.4) todos los valores de $V(s)$ utilizarán el valor pasivo con el índice λ. El sistema de ecuaciones es:

$$\lambda(3) = R_1(3) + \gamma\left(\frac{1}{2}V(3) + \frac{1}{2}V(4)\right) - R_0(3) - \gamma\left(\frac{1}{2}V(3) + \frac{1}{2}V(2)\right) = \frac{\gamma}{2}(V(4) - V(2))$$

$$\begin{aligned} V(1) &= R_0(1) + \lambda + \gamma\left[\frac{1}{2}V(1) + \frac{1}{2}V(4)\right] \\ V(2) &= R_0(2) + \lambda + \gamma\left[\frac{1}{2}V(2) + \frac{1}{2}V(1)\right] \\ V(3) &= R_0(3) + \lambda + \gamma\left[\frac{1}{2}V(3) + \frac{1}{2}V(2)\right] \\ V(4) &= R_0(4) + \lambda + \gamma\left[\frac{1}{2}V(4) + \frac{1}{2}V(3)\right] \end{aligned} \tag{3.1}$$

Este sistema de ecuaciones se puede resolver fácilmente expresándolo como una ecuación matricial de la forma:

$$\begin{aligned} \begin{pmatrix} V1 \\ V2 \\ V3 \\ V4 \end{pmatrix} &= \begin{pmatrix} \gamma/2 & 0 & 0 & \gamma/2 \\ \gamma/2 & \gamma/2 & 0 & 0 \\ 0 & \gamma/2 & \gamma/2 & 0 \\ 0 & 0 & \gamma/2 & \gamma/2 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \\ V4 \end{pmatrix} + \begin{pmatrix} -1+\lambda \\ \lambda \\ \lambda \\ 1+\lambda \end{pmatrix} \\ \begin{pmatrix} 1-\lambda \\ -\lambda \\ -\lambda \\ -1-\lambda \end{pmatrix} &= \begin{pmatrix} \gamma/2-1 & 0 & 0 & \gamma/2 \\ \gamma/2 & \gamma/2-1 & 0 & 0 \\ 0 & \gamma/2 & \gamma/2-1 & 0 \\ 0 & 0 & \gamma/2 & \gamma/2-1 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \\ V4 \end{pmatrix} \end{aligned} \tag{3.2}$$

Resolviendo para $V1, V2, V3$ y $V4$, y despejando en el término $\lambda(3)$ de la ecuación (3.1) obtenemos el valor del índice de Whittle para el estado 3 en función del término de descuento γ:

$$\lambda(3) = \frac{\gamma}{\gamma^2 - 2\gamma + 2} = g(3) \tag{3.3}$$

- **Estado 2**: Ahora el estado 3 se encuentra activo, mientras que el resto de estados siguen pasivos. El sistema de ecuaciones de nuestro sistema es ahora:

$$\lambda(2) = R_1(2) + \gamma\left(\frac{1}{2}V(2) + \frac{1}{2}V(3)\right) - R_0(2) - \gamma\left(\frac{1}{2}V(2) + \frac{1}{2}V(1)\right) = \frac{\gamma}{2}(V(3) - V(1))$$

$$\begin{aligned} V(1) &= R_0(1) + \lambda + \gamma\left[\frac{1}{2}V(1) + \frac{1}{2}V(4)\right] \\ V(2) &= R_0(2) + \lambda + \gamma\left[\frac{1}{2}V(2) + \frac{1}{2}V(1)\right] \\ V(3) &= R_1(3) + \gamma\left[\frac{1}{2}V(3) + \frac{1}{2}V(4)\right] \\ V(4) &= R_0(4) + \lambda + \gamma\left[\frac{1}{2}V(4) + \frac{1}{2}V(3)\right] \end{aligned} \tag{3.4}$$

Al igual que en el caso anterior, resolviendo este sistema de ecuaciones nos da el valor del índice de Whittle del estado 2 en función del descuento γ:

$$\lambda(2) = \frac{\gamma}{2} = g(2) \tag{3.5}$$

- **Estado 1**: Ahora se encuentran activos los estados 2 y 3, mientras que siguen pasivos los estados 1 y 4. El sistema de ecuaciones es:

$$\lambda(1) = R_1(1) + \gamma\left(\frac{1}{2}V(1) + \frac{1}{2}V(2)\right) - R_0(1) - \gamma\left(\frac{1}{2}V(1) - \frac{1}{2}V(4)\right) = \frac{\gamma}{2}(V(2) - V(4))$$

$$\begin{aligned} V(1) &= R_0(1) + \lambda + \gamma\left[\frac{1}{2}V(1) + \frac{1}{2}V(4)\right] \\ V(2) &= R_1(2) + \gamma\left[\frac{1}{2}V(2) + \frac{1}{2}V(3)\right] \\ V(3) &= R_1(3) + \gamma\left[\frac{1}{2}V(3) + \frac{1}{2}V(4)\right] \\ V(4) &= R_0(4) + \lambda + \gamma\left[\frac{1}{2}V(4) + \frac{1}{2}V(3)\right] \end{aligned} \tag{3.6}$$

Resolviendo para los valores de $V1, V2, V3$ y $V4$ y despejando en $\lambda(1)$, obtenemos:

$$\lambda(1) = \frac{-\gamma}{2} = g(1) \tag{3.7}$$

- **Estado 4**: Todos los estados se encuentran activos menos el estado 4. El sistema de ecuaciones que describe este sistema es:

$$\lambda(4) = R_1(4) + \gamma\left(\frac{1}{2}V(4) + \frac{1}{2}V(1)\right) - R_0(4) - \gamma\left(\frac{1}{2}V(4) + \frac{1}{2}V(3)\right) = \frac{\gamma}{2}(V(1) - V(3))$$

$$\begin{aligned} V(1) &= R_1(1) + \gamma\left[\frac{1}{2}V(1) + \frac{1}{2}V(2)\right] \\ V(2) &= R_1(2) + \gamma\left[\frac{1}{2}V(2) + \frac{1}{2}V(3)\right] \\ V(3) &= R_1(3) + \gamma\left[\frac{1}{2}V(3) + \frac{1}{2}V(4)\right] \\ V(4) &= R_0(4) + \lambda + \gamma\left[\frac{1}{2}V(4) + \frac{1}{2}V(3)\right] \end{aligned} \tag{3.8}$$

Resolviendo este sistema de ecuaciones, obtenemos el índice del estado 4 en función del factor de descuento γ:

$$\lambda(4) = \frac{-\gamma}{\gamma^2 - 2\gamma + 2} = g(4) \tag{3.9}$$

Si realizamos el límite de los índices para $\gamma \to \infty$, vemos como todos estos convergen a los valores calculados en [32].

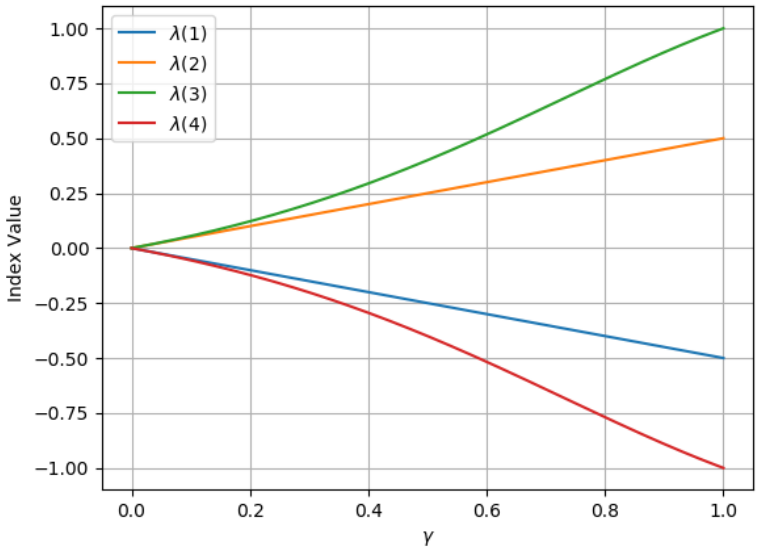


Figura 3.2: Valor de los índices de Whittle para el modelo de dinámica circular en función del factor de descuento γ

3.1.2. Problema con reinicio

Para el siguiente caso, consideraremos un espacio de estados $S = \{1,2,3,4,5\}$ con dos posibles acciones: pasivo ($u = 0$), donde el bandido tendrá un 90 % de probabilidades de avanzar un estado y un 10 % de permanecer en el mismo y activo ($u = 1$), donde el bandido vuelve al estado inicial con probabilidad 1. Las matrices de transición de estos procesos son:

$$P_0 = \begin{pmatrix} 1/10 & 9/10 & 0 & 0 & 0 \\ 1/10 & 0 & 9/10 & 0 & 0 \\ 1/10 & 0 & 0 & 9/10 & 0 \\ 1/10 & 0 & 0 & 0 & 9/10 \\ 1/10 & 0 & 0 & 0 & 9/10 \end{pmatrix} \qquad P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

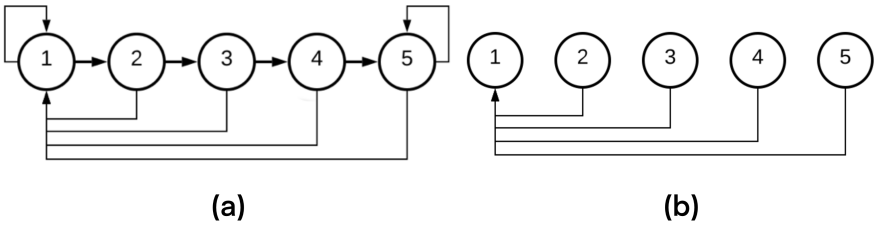


Figura 3.3: Visualización de la dinámica de los bandidos en el problema del reinicio, para la acción activa (a) y pasiva (b)

A diferencia del caso anterior, consideraremos recompensas diferentes en función de la acción que tomemos. De esta forma, si un bandido pasa a estar activo, su recompensa siempre será 0, mientras que si es pasivo su recompensa será $R_0(k) = a^k$, con k el número del estado, y donde tomaremos $a = 0{,}9$. Este problema tiene un especial interés ya que los estados más elevados son mucho menos visitados durante el entrenamiento del algoritmo, lo cual dificulta el proceso de aprendizaje.
Los índices de Whittle de este sistema, empleando el criterio de recompensa media en el horizonte infinito (ecuación (1.6)) han sido estudiados previamente en [33], obteniendo los valores $g(1) = -0{,}9, g(2) = -0{,}73, g(3) = -0{,}5, g(4) = -0{,}26$ y $g(5) = -0{,}01$. Con nuestro criterio de "optimalidad", los valores de los índices dependerán del factor de descuento γ, aunque el orden de activación de los estados será el mismo. Por lo tanto, igual que en el apartado 3.1.1, calcularemos el valor teórico de estos índices empleando las ecuaciones (2.4) y (2.9) activando los bandidos en el orden $5 \to 4 \to 3 \to 2 \to 1$.

- **Estado 5**: Todos los estados se encuentran pasivos. El sistema de ecuaciones de este caso es:

$$\lambda(5) = R_1(5) + \gamma V(1) - R_0(5) - \gamma\left(\frac{9}{10}V(5) + \frac{1}{10}V(1)\right) = -(0{,}9)^5 + \frac{9}{10}\gamma(V(1) - V(5))$$

$$\begin{aligned} V(1) &= R_0(1) + \lambda + \gamma\left[\frac{9}{10}V(2) + \frac{1}{10}V(1)\right] \\ V(2) &= R_0(2) + \lambda + \gamma\left[\frac{9}{10}V(3) + \frac{1}{10}V(1)\right] \\ V(3) &= R_0(3) + \lambda + \gamma\left[\frac{9}{10}V(4) + \frac{1}{10}V(1)\right] \\ V(4) &= R_0(4) + \lambda + \gamma\left[\frac{9}{10}V(5) + \frac{1}{10}V(1)\right] \\ V(5) &= R_0(5) + \lambda + \gamma\left[\frac{9}{10}V(5) + \frac{1}{10}V(1)\right] \end{aligned} \tag{3.10}$$

Al igual que en la sección 3.1.1, este problema se puede resolver planteándolo como un sistema

matricial, en el que calcularemos los valores $V1, V2, V3, V4$ y $V5$ y los despejaremos en el término $\lambda(5)$ de la ecuación anterior:

$$\begin{pmatrix} V(1) \\ V(2) \\ V(3) \\ V(4) \\ V(5) \end{pmatrix} = \begin{pmatrix} \gamma/10 & 9\gamma/10 & 0 & 0 & 0 \\ \gamma/10 & 0 & 9\gamma/10 & 0 & 0 \\ \gamma/10 & 0 & 0 & 9\gamma/10 & 0 \\ \gamma/10 & 0 & 0 & 0 & 9\gamma/10 \\ \gamma/10 & 0 & 0 & 0 & 9\gamma/10 \end{pmatrix} \begin{pmatrix} V(1) \\ V(2) \\ V(3) \\ V(4) \\ V(5) \end{pmatrix} + \begin{pmatrix} R_0(1)+\lambda \\ R_0(2)+\lambda \\ R_0(3)+\lambda \\ R_0(4)+\lambda \\ R_0(5)+\lambda \end{pmatrix}$$
$$\begin{pmatrix} -R_0(1)-\lambda \\ -R_0(2)-\lambda \\ -R_0(3)-\lambda \\ -R_0(4)-\lambda \\ -R_0(5)-\lambda \end{pmatrix} = \begin{pmatrix} \gamma/10-1 & 9\gamma/10 & 0 & 0 & 0 \\ \gamma/10 & -1 & 9\gamma/10 & 0 & 0 \\ \gamma/10 & 0 & -1 & 9\gamma/10 & 0 \\ \gamma/10 & 0 & 0 & -1 & 9\gamma/10 \\ \gamma/10 & 0 & 0 & 0 & 9\gamma/10-1 \end{pmatrix} \begin{pmatrix} V(1) \\ V(2) \\ V(3) \\ V(4) \\ V(5) \end{pmatrix} \tag{3.11}$$

El índice para el estado 5 en función del factor de descuento γ es:

$$\lambda(5) = \frac{43046721\gamma^4}{1000000000} + \frac{10097379\gamma^3}{100000000} + \frac{1778031\gamma^2}{10000000} + \frac{278559\gamma}{1000000} - \frac{59049}{100000} = g(5) \tag{3.12}$$

- **Estado 4**: Todos los estados menos el 5 se encuentran pasivos. El sistema de ecuaciones ahora es:

$$\lambda(4) = R_1(4) + \gamma V(1) - R_0(4) - \gamma\left(\frac{9}{10}V(5) - \frac{1}{10}V(1)\right) = -(0{,}9)^4 + \frac{9}{10}\gamma(V(1) - V(5))$$

$$\begin{aligned} V(1) &= R_0(1) + \lambda + \gamma\left[\frac{9}{10}V(2) + \frac{1}{10}V(1)\right] \\ V(2) &= R_0(2) + \lambda + \gamma\left[\frac{9}{10}V(3) + \frac{1}{10}V(1)\right] \\ V(3) &= R_0(3) + \lambda + \gamma\left[\frac{9}{10}V(4) + \frac{1}{10}V(1)\right] \\ V(4) &= R_0(4) + \lambda + \gamma\left[\frac{9}{10}V(5) + \frac{1}{10}V(1)\right] \\ V(5) &= R_1(5) + \gamma V(1) \end{aligned} \tag{3.13}$$

Resolviendo para los $V1, V2, V3, V4$ y $V5$ y despejando en $\lambda(4)$ obtenemos el valor del índice de Whittle para el estado 4 en función del factor de descuento γ:

$$\lambda(4) = \frac{81\left(6561\gamma^3 + 15390\gamma^2 + 27100\gamma - 81000\right)}{10000000} = g(4) \tag{3.14}$$

- **Estado 3**: Los estados 4 y 5 se encuentran activos mientras que los estados 1, 2 y 3 se mantienen pasivos. El sistema de ecuaciones es:

$$\lambda(3) = R_1(3) + \gamma V(1) - R_0(3) - \gamma\left(\frac{9}{10}V(4) - \frac{1}{10}V(1)\right) = -(0{,}9)^3 + \frac{9}{10}\gamma(V(1) - V(4))$$

$$\begin{aligned} V(1) &= R_0(1) + \lambda + \gamma\left[\frac{9}{10}V(2) + \frac{1}{10}V(1)\right] \\ V(2) &= R_0(2) + \lambda + \gamma\left[\frac{9}{10}V(3) + \frac{1}{10}V(1)\right] \\ V(3) &= R_0(3) + \lambda + \gamma\left[\frac{9}{10}V(4) + \frac{1}{10}V(1)\right] \\ V(4) &= R_1(4) + \gamma V(1) \\ V(5) &= R_1(5) + \gamma V(1) \end{aligned} \tag{3.15}$$

Resolviendo este sistema de ecuaciones, obtenemos el índice del tercer estado en función de γ:

$$\lambda(3) = \frac{6561\gamma^2}{100000} + \frac{1539\gamma}{10000} - \frac{729}{1000} = g(3) \tag{3.16}$$

- **Estado 2**: Todos los estados menos el 1 y el 2 están activos. El sistema de ecuaciones es:

$$\lambda(2) = R_1(2) + \gamma V(1) - R_0(2) - \gamma\left(\frac{9}{10}V(3) - \frac{1}{10}V(1)\right) = -(0{,}9)^2 + \frac{9}{10}\gamma(V(1) - V(3))$$

$$\begin{aligned} V(1) &= R_0(1) + \lambda + \gamma\left[\frac{9}{10}V(2) + \frac{1}{10}V(1)\right] \\ V(2) &= R_0(2) + \lambda + \gamma\left[\frac{9}{10}V(3) + \frac{1}{10}V(1)\right] \\ V(3) &= R_1(3) + \gamma V(1) \\ V(4) &= R_1(4) + \gamma V(1) \\ V(5) &= R_1(5) + \gamma V(1) \end{aligned} \tag{3.17}$$

Cuyo índice de Whittle en función de γ es:

$$\lambda(2) = \frac{81\gamma}{1000} - \frac{81}{100} = g(2) \tag{3.18}$$

- **Estado 1**: Por último, nos encontramos con el caso de que todos los estados menos el 1 estén activos. el sistema de ecuaciones ahora es:

$$\lambda(1) = R_1(1) + \gamma V(1) - R_0(1) - \gamma\left(\frac{9}{10}V(2) - \frac{1}{10}V(1)\right) = -0{,}9 + \frac{9}{10}\gamma(V(1) - V(2))$$

$$\begin{aligned} V(1) &= R_0(1) + \lambda + \gamma\left[\frac{9}{10}V(2) + \frac{1}{10}V(1)\right] \\ V(2) &= R_1(2) + \gamma V(1) \\ V(3) &= R_1(3) + \gamma V(1) \\ V(4) &= R_1(4) + \gamma V(1) \\ V(5) &= R_1(5) + \gamma V(1) \end{aligned} \tag{3.19}$$

En este caso, el índice de Whittle para este estado es constante y por lo tanto no depende del factor de descuento γ:

$$\lambda(1) = -\frac{9}{10} = g(1) \tag{3.20}$$

Igual que antes, nuestros resultados convergen con los de [33] en el límite $\gamma \to 1$.

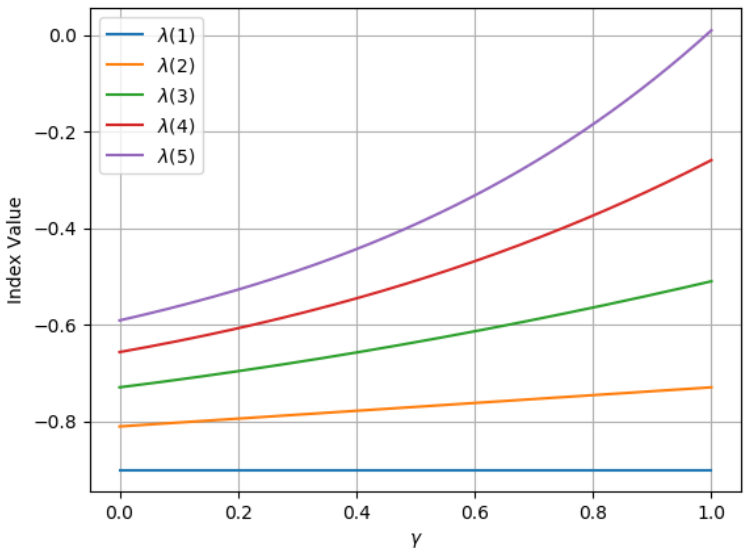


Figura 3.4: Valor de los índices de Whittle para el problema con reinicio en función del factor de descuento γ

Una vez que hemos visto los valores teóricos a los que tienen que converger los índices de Whittle para el problema de la dinámica circular (sección 3.1.1) y el problema con reinicio (sección 3.1.2), vamos a pasar a discutir el desarrollo del algoritmo a partir del cual realizaremos el proceso de Q-learning y el cálculo de los índices de Whittle en escalas de tiempo separadas.

3.2. Esquema del algoritmo

Una vez hemos visto los valores teóricos de los índices de Whittle para los casos de dinámica circular y el problema de reinicio, vamos a pasar a estudiar la estructura del algoritmo y como el proceso de Q-learning expuesto en la sección 2.3.2.

Para el cálculo de los índices, consideraremos en ambos casos $N = 100$ bandidos distintos, de los cuales solo están activos en todo momento $K = 20$ bandidos. En ambos problemas, inicializaremos la Q-table como un conjunto de tablas, una para cada bandido, de dimesión $S \times A \times S$, donde S es el número de estados y A el número de acciones posibles (en este caso, siempre será 2). El primer término S hace referencia a los estados x en los que el índice $g(x)$ hace que estar pasivo o activo sea igualmente deseable, mientras que el segundo término S hace referencia al estado visitado por el proceso de Q-learning, s_t. Al empezar el algoritmo, los índices de Whittle para cada estado están inicializados a 0.

La secuencia de control está definida a través del algoritmo de "Epsilon-greedy": Sea ε un parámetro de valor $0 < \varepsilon < 1$, tomamos con probabilidad $1-\varepsilon$ los K bandidos con el índice $g(x)$ más alto y los definimos como *activos*, mientras que el resto se mantienen *pasivos*, es decir, tomamos una política *greedy* en la que elegimos la opción más eficiente: **explotamos** nuestro conocimiento sobre el problema. Por otro lado, con probabilidad ε, se seleccionan K bandidos al azar para establecerlos como activos. Este tipo de acciones pueden ser menos eficaces que aquellas de la política *greedy*, pero también pueden dar lugar al descubrimiento de políticas más óptimas que las anteriores: **exploramos** la dinámica de este problema, aumentando nuestro conocimiento sobre el mismo. Este tipo de secuencia de control permite un buen balance entre *exploración* y *explotación* de la información del problema. En nuestro caso, utilizaremos $\varepsilon = 0{,}1$, de modo que un 10 % de las veces exploraremos al azar para obtener nueva información sobre el problema mientras que el 90 % de las veces tomaremos una decisión acorde a la política que estamos construyendo.

Bucle para cada iteración n:
- Seleccionar a través del algoritmo ε−greedy los bandidos a activar
- Simular los nuevos estados para todos los bandidos en función de la dinámica del problema
- Calcular la recompensa de estos estados
- Calcular los nuevos tamaños de paso $\alpha(n)$ y $\beta(n)$:
  - $\alpha(n) = \frac{1}{\lceil \frac{n}{500} \rceil}$
  - $\beta(n) = \frac{1}{1+\lceil \frac{n \log n}{500} \rceil} I\{n(\text{mod}N) \equiv 0\}$
- Calcular los Q−values de la Q−table de cada bandido:
  - $Q^x_{n+1}(s_t, a_t) \leftarrow$
    $Q^x_n(s_t, a_t) + \alpha(n)\left[(1-a_t)(R_0(s_t) + g_n(x)) + a_t R_1(s_t) + \gamma \text{máx}_{v\in\{0,1\}} Q^x_n(s_{t+1}, v) - Q^x_n(s_t, a_t)\right]$
- Calcular los nuevos índices de Whittle:
  - $g_{n+1}(x) = g_n(x) + \beta(n)\left(Q^x_n(x,1) - Q^x_n(x,0)\right)$
- Actualizar los valores de los estados

Figura 3.5: Esquema del algoritmo de aprendizaje

Donde los valores α y β están definidos en (2.13) y el cálculo de los Q-values y los índices son los descritos en (2.11) y (2.12) respectivamente.

En la definición de β de (2.13b) empleamos el término $I\{n(\text{mod}N) \equiv 0\}$. Esto implica que β es distinto de 0 solo si el número de la iteración n es un múltiplo del número de bandidos N. En la figura 3.5, en cada iteración actualizamos los índices de Whittle (2.12), al igual que los Q-values (2.11). Sin embargo, solo cuando n es múltiplo de N hay un cambio real en el valor de los índices de Whittle, ya que en el resto de casos $g_{n+1}(x) = g_n(x)$. Es así que, aunque actualicemos ambos términos en todas las iteraciones, solo los Q-values cambian en todas ellas. De este modo, obtenemos el efecto de las dos

escalas de tiempo, en el que los Q-values se actualizan más a menudo que los índices.

3.2.1. Demostración de la convergencia

Nuestro algoritmo se apoya en la convergencia, por un lado, del modelo de Q-learning clásico introducido en 1.3.6, y por otro del esquema de dos escalas temporales planteados en [34] y [35]. A continuación, mostraremos un boceto de la demostración de la convergencia de ambos sistemas.

Convergencia de Q-learning

Para la demostración de la convergencia del algoritmo de Q-learning, consideremos un Proceso de Decisión de Markov con la tupla (S,A,P,R) donde S es el espacio finito de estados de la cadena, A es el espacio finito de acciones, P son las probabilidades de transmisión y R es la función de recompensa. Denotaremos los elementos de S como x e y y los elementos de A como a y b. La función de recompensa por tanto esta definida a través del triplete (x,a,y):

$$r : S \times A \times S \to \mathbb{R}$$

Donde obtenemos una recompensa $R(x,a,y)$ por cada transición del estado x al estado y al realizar una acción a.

Partiendo del criterio de optimalidad de horizonte infinito descontado con el que hemos estado empleando, la función de valor acción-estado, bajo una secuencia de controles $\{A_t\}$ es

$$Q(x,\{A_t\}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t,A_t)|X_0 = x\right]$$

Por lo tanto, para cada estado $x \in S$, la función de valor de estado óptima está definida como:

$$V^*(x) = \max_{A_t} Q(x,\{A_t\})$$

La cual verifica

$$V^*(x) = \max_{a\in A} \sum_{y\in X} P_a(x,a)\left[R(x,a,y) + \gamma V^*(y)\right]$$

Y por lo tanto, la función de valor acción-estado óptima es

$$Q^*(x,a) = \sum_{y\in X} P_a(x,y)\left[R(x,a,y) + \gamma V^*(y)\right]$$

Esta función de valor óptima es un punto fijo del operador de contracción **H** definido para una función genérica $q : S \times A \to \mathbb{R}$ como:

$$(\mathbf{H}q)(x,a) = \sum_{y\in S} P_a(x,y)\left[R(x,a,y) + \gamma \max_{b\in A} q(y,b)\right] \tag{3.21}$$

Este operador es una contracción en la norma uniforme, es decir,

$$\|\mathbf{H}q_1 - \mathbf{H}q_2\|_\infty \leq \gamma \|q_1 - q_2\|_\infty \tag{3.22}$$

Delimitando así el valor de la función genérica q. La demostración de esta desigualdad se puede obtener

desarrollando la ecuación anterior con la definición (3.21):

$$\begin{aligned}\|\mathbf{H}q_1 - \mathbf{H}q_2\|_\infty &= \max_{x,a}\left|\sum_{y\in X} P_a(x,y)\left[R(x,a,y)+\gamma\max_{b\in A} q_1(y,b) - R(x,a,y) - \gamma\max_{b\in A} q_2(y,b)\right]\right| = \\ &= \max_{x,a}\gamma\left|\sum_{y\in S} P_a(x,y)\left[\max_{b\in A} q_1(y,b) - \max_{b\in A} q_2(y,b)\right]\right| \leq \\ &\leq \max_{x,a}\gamma\sum_{y\in S} P_a(x,y)\left|\max_{b\in A} q_1(y,b) - \max_{b\in A} q_2(y,b)\right| \leq \\ &\leq \max_{x,a}\gamma\sum_{y\in S} P_a(x,y)\max_{z,b}|q_1(z,b) - q_2(z,b)| = \\ &= \max_{x,a}\gamma\sum_{y\in S} P_a(x,y)\|q_1 - q_2\|_\infty = \gamma\|q_1 - q_2\|_\infty\end{aligned}$$

El algoritmo de Q-learning determina la función óptima de valor empleando distintas muestras durante el aprendizaje. Consideremos una política aleatoria π tal que la probabilidad de realizar una acción a bajo un estado x sea no nula, es decir

$$\mathbb{P}_\pi\left[A_t = a | X_t = x\right] > 0$$

para cualquier par estado-acción (x,a). Sea $\{x_t\}$ la secuencia de estados obtenida siguiendo la política π, $\{a_t\}$ la secuencia de acciones realizadas y $\{R_t\}$ las recompensas obtenidas, para cualquier estimación inicial Q_0, el algoritmo de actualización de Q-learning es

$$Q_{t+1}(x_t,a_t) = Q_t(x_t,a_t) + \alpha_t(x_t,a_t)\left[R_t + \gamma\max_{b\in A} Q_t(x_{t+1},b) - Q_t(x_t,a_t)\right]$$

Donde $\alpha_t(x_t,a_t)$ es el tamaño empleado en la iteración t para el par de estado-acción (x_t,a_t), el cual verifica $0 \leq \alpha_t(x,a) \leq 1$. Estas actualizaciones son asíncronas, es decir, en cada iteración no actualizamos todos los valores de $Q(x,a)$ para todos $x \in S$ y $a \in A$, sino solamente la tupla (x_t,a_t). Esto nos lleva a los siguientes teoremas.

- **Teorema 1.** Dado una MDP finita definida por la tupla (S,A,T,R), el algoritmo de Q-learning dado por

$$Q_{t+1}(x_t,a_t) = Q_t(x_t,a_t) + \alpha_t(x_t,a_t)\left[R_t + \gamma\max_{b\in A} Q_t(x_{t+1},b) - Q_t(x_t,a_t)\right] \tag{3.23}$$

  Converge al valor óptimo de la función de valor Q siempre y cuando

$$\sum_t \alpha_t(x,a) = \infty \qquad \sum_t \alpha_t^2(x,a) < \infty$$

  Para todos los $(x,a) \in S \times A$. Debido a que el valor de α está delimitado a $0 \leq \alpha_t(x,a) \leq 1$, la condición anterior requiere que todos los pares de estado-acción sean visitados. Antes de demostrar este teorema, debemos presentar un resultado auxiliar de la teoría de aproximación estocástica.

- **Teorema 2.** Sea un proceso aleatorio $\{\Delta_t\}$ que tome valores en $\mathbb{R}^n$ y esté definido como

$$\Delta_{t+1}(x) = (1 - \alpha_t(x))\Delta_t(x) + \alpha(x)F_t(x)$$

  Este converge a 0 bajo las siguientes condiciones

  - $0 \leq \alpha_t \leq 1, \sum_t \alpha_t(x) = \infty$ y $\sum_t \alpha_t^2(x) < \infty$

- $\|\mathbb{E}[F_t(x)|\mathscr{F}_t]\|_W \leq \gamma\|\Delta_t\|_W$, con $\gamma < 1$
- $\mathbf{var}[F_t(x)|\mathscr{F}_t] \leq C(1+\|\Delta_t\|_W^2)$, para $C > 0$.

La demostración de este segundo teorema se encuentra en [36].

- **Demostración del Teorema 1.** Empezaremos reescribiendo la ecuación (3.23) como

$$Q_{t+1}(x_t,a_t) = (1-\alpha_t(x_t,a_t))Q_t(x_t,a_t) + \alpha_t(x_t,a_t)\left[R_t + \gamma \max_{b\in A} Q_t(x_{t+1},b)\right]$$

Restando a ambos lados de la ecuación $Q^*(x_t,a_t)$ y definiendo $\Delta_t(x,a) = Q_t(x,a) - Q^*(x,a)$, tenemos

$$\Delta_t(x_t,a_t) = (1-\alpha_t(x_t,a_t))\Delta_t(x_t,a_t)) + \alpha_t(x,a)\left[R_t + \gamma \max_{b\in A} Q_t(x_{t+1},b) - Q^*(x_t,a_t)\right]$$

Definiendo $F_t(x,a) = R(x,a,X(x,a)) + \gamma \max_{b\in a} Q_t(y,b) - Q^*(x,a)$, donde $X(x,a)$ es una muestra aleatoria obtenida de la cadena de Markov con espacio de estado S y probabilidad de transmisión T_a, tenemos

$$\mathbb{E}[F_t(x,a)|\mathscr{F}_t] = \sum_{y\in S} T_a(x,y)\left[R(x,a,y) + \gamma \max_{b\in A} Q_t(y,b) - Q^*(x,a)\right] =$$
$$= (\mathbf{H}Q_t)(x,a) - Q^*(x,a)$$

De tal manera que, empleando la definición de $Q^* = \mathbf{H}Q^*$,

$$\mathbb{E}[F_t(x,a)|\mathscr{F}_t] = (\mathbf{H}Q_t)(x,a) - (\mathbf{H}Q^*)(x,a)$$

Empleando la definición de $\mathbf{H}$ en (3.22)

$$\|\mathbb{E}[F_t(x,a)|\mathscr{F}_t]\|_\infty \leq \gamma\|Q_t - Q^*\|_\infty = \gamma\|\Delta_t\|_\infty$$

Por último

$$\mathbf{var}[F_t(x)|\mathscr{F}_t] =$$
$$= \mathbb{E}\left[\left(R(x,a,X(x,a)) + \gamma \max_{b\in A} Q_t(y,b) - Q^*(x,a) - (\mathbf{H}Q_t)(x,a) + Q^*(x,a)\right)^2\right] =$$
$$= \mathbb{E}\left[\left(R(x,a,X(x,a)) + \gamma \max_{b\in A} Q_t(y,b) - (\mathbf{H}Q_t)(x,a)\right)^2\right] =$$
$$= \mathbf{var}\left[R(x,a,X(x,a)) + \gamma \max_{b\in A} Q_t(y,b)|\mathscr{F}_t\right]$$

Dado que la función de recompensa R está delimitada, el resultado anterior verifica

$$\mathbf{var}[F_t(x)|\mathscr{F}_t] \leq C(1+\|\Delta_t\|_W^2)$$

Para una constante C. A través del Teorema 2, Δ_t converge a cero y por lo tanto Q_t converge al valor óptimo, Q^*.

Convergencia de las dos escalas de tiempo

Para la demostración de la estabilidad de nuestro sistema con dos escalas de tiempo, seguiremos la demostración propuesta en [34] y [35]. Primero, generalizaremos las expresiones (2.11) y (2.12) para el cálculo de los Q-value y los índices de Whittle en las siguientes expresiones:

$$x_{n+1} = x_n + a(n)\left[h(x_n, y_n) + M_{n+1}^{(1)}\right] \tag{3.24a}$$

$$y_{n+1} = y_n + b(n)\left[g(x_n, y_n) + M_{n+1}^{(2)}\right] \tag{3.24b}$$

Donde (3.24a) es la escala rápida, que representa el cálculo del Q-value (2.11) y (3.24b) es la escala lenta, que representa los índices de Whittle en la ecuación (2.12). En estas ecuaciones, h y g son funciones Lipschitz (continuas), M_n son secuencias de diferencias Martingale, que representan términos de ruido, y $a(n)$ y $b(n)$ son términos de tamaño de paso que disminuyen tal que $\frac{b(n)}{a(n)} \to 0$ cuando $n \to \infty$. y_n dicta el comportamiento del agente, definiendo su política, mientras que x_n es un valor acumulado que ayudará a definir y_n. Nuestro objetivo es demostrar que $y_n \to y^*$ y $x_n \to x^*$, donde y^* y x^* son aquellos para los cuales se obtiene esta política óptima. En esta demostración, es importante asegurar que tanto x_n como y_n son estables, es decir, $\sup_n ||x_n|| < \infty$ y $\sup_n ||y_n|| < \infty$. Aunque podríamos realizar una proyección de x_n e y_n en un subconjunto $\mathbf{C}$ que los haga estables, esto podría excluir los términos x^*, y^* de $\mathbf{C}$; de ahí la importancia de asegurar esta estabilidad en los valores originales.

Primero, definiremos $F_{su}^{\lambda}(\Psi(j,b))$ y $M_{n+1}(s,u)$ tal que:

$$F_{su}^{\lambda}(\Psi(j,b)) = (1-u)(R_0(s)+\lambda) + uR_1(s) + \gamma \sum_j p(j|i,u) \max_{v\in\{0,1\}} \Psi(j,v)$$

$$M_{n+1}(s,u) = (1-u)(R_0(s)+\lambda_n(x)) + uR_1(s) + \max_{v\in\{0,1\}} Q_n(x_{n+1},v) - F_{su}^{\lambda_n(x)}(Q_n)$$

A partir de estos términos, podemos reescribir la ecuación (2.11) como:

$$Q_{n+1}^x(s,u) = Q_n^x(i,u) + \alpha(n)\left[F_{su}^{\lambda_n(x)}(Q_n) - Q_n + M_{n+1}(s,u)\right] \tag{3.25}$$

Si comparamos las ecuaciones (3.25) y (3.24a), vemos como $a(n) = \alpha(n)$, $h(x_n, y_n) = F_{su}^{\lambda_n(x)}(Q_n) - Q_n$ donde $x_n = Q_n$ e $y_n = g_n$ son el Q-value y el índice de Whittle respectivamente y $M_{n+1}(s,u)$ es la secuencia diferencial Martingale $M_{n+1}^{(1)}$.

Por otro lado, comparando las ecuaciones (2.12) y (3.24b), vemos como $b(n) = \beta(n)$, $g(x_n, y_n) = Q_n^x(x,1) - Q_n^x(x,0)$ y la secuencia Martingale $M_{n+1}^{(2)} = 0$.

En [35] citan 3 condiciones necesarias para que las ecuaciones (3.24) puedan ser estables y converger

A1 h y g deben ser funciones Lipschitz continuas.

A2 $\{M_n^{(1)}\}$ y $\{M_n^{(2)}\}$ son secuencias diferenciales Martingale.

A3 $\{a(n)\}$ y $\{b(n)\}$ satisfacen:

- $a(n) > 0, b(n) > 0$
- $\sum_n a(n) = \sum_n b(n) = \infty$, $\sum_n \left(a(n)^2 + b(n)^2\right) < \infty$
- $\frac{b(n)}{a(n)} \to \infty$

La demostración de la condición **A1** se encuentra detallada en la página 687 de [37]. En nuestra notación, $M_{n+1}(s,u)$ y 0 son respectivamente las secuencias de diferencias Martingale $M_{n+1}^{(1)}$ y $M_{n+1}^{(2)}$, cumpliendo así la condición **A2**. **A3** también está verificada a partir de la defínción de $\alpha(n)$ y $\beta(n)$ en (2.13a) y (2.13b).

En la demostración de la convergencia del índice λ al índice de Whittle g_n, vamos a considerar primero que las ecuaciones (2.11) y (2.12) están delimitadas. Más tarde demostraremos esta condición.

Primero, reescribiremos la ecuación para el cálculo de los índices de Whittle (2.12) como:

$$g_{n+1}(x) = g_n(x) + \alpha(n)\left(\frac{\beta(n)}{\alpha(n)}\right)\left(Q_n^x(x,1) - Q_n^x(x,0)\right) \tag{3.26}$$

Sea $\tau(n) = \sum_{m=0}^{n} \alpha(n)$, definimos la interpolación:

$$\bar{Q}(t) = Q(n) + \left(\frac{t-\tau(n)}{\tau(n+1)-\tau(n)}\right)(Q(n+1) - Q(n)) \tag{3.27a}$$

$$\bar{g}(t) = g(n) + \left(\frac{t-\tau(n)}{\tau(n+1)-\tau(n)}\right)(g(n+1) - g(n)) \tag{3.27b}$$

$$t \in [\tau(n), \tau(n+1)]$$

Estas trayectorias siguen el comportamiento de las ODE's delimitadas

$$\dot{Q}(t) = h(Q(t), g(t)), \qquad \dot{g}(t) = 0$$

Donde $\dot{g}(t) = 0$ debido a que $\frac{\beta(n)}{\alpha(n)} \to 0$ a medida que $n \to \infty$. Desde el sistema de referencia de $Q(t)$, g es constante, con valor g'. Gracias a esto, la primera ODE pasa a ser $\dot{Q} = h(Q(t), g')$, la cual al estar bien definida y delimitada, posee un equilibrio asintóticamente estable en Q_λ^* (teorema 3.4 en la página 689, [37]). Esto implica que, a medida que aumentemos n, $Q_n^x - Q_{\lambda_n}^* \to 0$. Por otro lado, el caso de $g(t)$, consideramos una segunda trayectoria, en otra escala temporal tal que:

$$\tilde{g}(t) = g(n) + \left(\frac{t-\tau'(n)}{\tau'(n+1) - tau'(n)}\right)(g(n+1) - g(n)),$$
$$t \in [\tau(n), \tau'(n+1)], \qquad \tau'(n) = \sum_{m=0}^{n} \beta(m), n \geq 0 \tag{3.28}$$

Esta trayectoria seguirá la ODE

$$\dot{\Lambda}(t) = Q_{\Lambda(t)}^*(x,1) - Q_{\Lambda(t)}^*(x,0)$$

Si $\Lambda(t)$ es mayor que el índice de Whittle óptimo para un estado dado, $g^*(x)$, tendremos un exceso de subsidio, en el que se preferirá la acción pasiva a la activa y $\dot{\Lambda}(t) < 0$, de modo que $\Lambda(t)$ decrecerá. De la misma manera, si $\Lambda(t) < g^*(x)$, quiere decir que no estamos considerando suficiente subsidio, y siempre se preferirá la acción activa, de modo que $\dot{\Lambda}(t) > 0$ y $\Lambda(t)$ crecerá: la trayectoria de $\Lambda(t)$ queda así delimitada. Igual que en el caso anterior, al ser una ODE bien definida y delimitada, existe un punto de equilibrio asintóticamente estable al que converge, en el que Λ satisface $Q_\Lambda^*(x,1) = Q_\Lambda^*(x,0)$ y ambas políticas son igualmente deseables, es decir, Λ es el índice de Whittle $g(x)$.

3.3. Resultados numéricos

En la sección 3.1 planteamos dos problemas con dinámicas distintas: por un lado, un sistema con dinámica circular, donde el último elemento de una cadena de estados conecta con el primero, con unas matrices de transmisión

$$P_0 = \begin{pmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix} \qquad P_1 = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \end{pmatrix}$$

Por otro, un problema con reinicio, en el que una acción *pasiva* hace avanzar un estado en la cadena de Markov (con una probabilidad del 90 %), mientras que una acción activa hace volver al inicio de la cadena, con matrices de transmisión

$$P_0 = \begin{pmatrix} 1/10 & 9/10 & 0 & 0 & 0 \\ 1/10 & 0 & 9/10 & 0 & 0 \\ 1/10 & 0 & 0 & 9/10 & 0 \\ 1/10 & 0 & 0 & 0 & 9/10 \\ 1/10 & 0 & 0 & 0 & 9/10 \end{pmatrix} \qquad P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ambos sistemas tienen espacios de estados distintos, 4 y 5 estados para cada problema respectivamente, y recompensas diferentes. A su vez, ambos problemas se pueden modificar fácilmente para contemplar más estados y distintos tipos de recompensas.

3.3.1. Dinámica circular

Tal y como hemos descrito en la sección 3.1.1, planteamos un sistema de 4 estados en los que las recompensas no dependen de la acción tomada en cada uno de ellos, y vienen definidas como $R(1) = -1, R(2) = 0, R(3) = 0, R(4) = 1$. Los valores teóricos de los índices de Whittle de estos estados son:

$$g(1) = \frac{-\gamma}{2} \tag{3.29a}$$

$$g(2) = \frac{\gamma}{2} \tag{3.29b}$$

$$g(3) = \frac{\gamma}{\gamma^2 - 2\gamma + 2} \tag{3.29c}$$

$$g(4) = \frac{-\gamma}{\gamma^2 - 2\gamma + 2} \tag{3.29d}$$

Como podemos ver en la gráfica 3.2, cuanto menor es el valor de γ, más parecidos son los índices entre sí, hasta acabar convergiendo en el origen. En nuestro estudio, hemos entrenado el algoritmo descrito en la sección 3.2 con 100000 iteraciones para distintos valores de γ entre 0.05 y 0.90. Para cada valor del parámetro de descuento, hemos comparado el error numérico del índice con respecto a su valor teórico en la ecuación (3.29). Los valores absolutos de estos errores para cada estado, junto con un promedio, se encuentran en la tabla 3.1 y la gráfica 3.6. A medida que aumentamos el valor del parámetro γ y nos acercamos a 1, el error numérico aumenta cada vez más. Este tipo de errores es común en el criterio de optimalidad de horizonte infinito descontado [38] [39], donde al aumentar el valor de γ y acercarlo a 1, la suma en la ecuación (1.5) diverge. Incluso con un factor de descuento ligeramente menor que 1, el aprendizaje de los Q-values lleva a la propagación de errores e inestabilidades [39]. Para minimizar el error numérico en los índices de Whittle y al mismo tiempo darle peso a los estados

en futuros en el cálculo de los Q-values, emplearemos $\gamma = 0{,}3$ que, como podemos ver en la gráfica 3.6, tiene uno de los errores promedio más bajos. Emplearemos este valor para discutir el resto de los resultados para este problema.

γ	Estado 1	Estado 2	Estado 3	Estado 4	Promedio
0,05	3,19E-03	9,79E-04	1,97E-03	3,51E-02	1,03E-02
0,1	1,75E-04	1,94E-03	2,45E-03	1,80E-02	5,65E-03
0,15	3,44E-03	2,40E-03	3,77E-03	3,48E-03	3,27E-03
0,2	6,01E-03	2,03E-03	7,26E-03	6,14E-03	5,36E-03
0,25	7,45E-03	1,60E-03	7,97E-03	9,89E-03	6,73E-03
0,3	4,76E-03	6,38E-04	1,12E-02	5,20E-03	5,44E-03
0,35	8,93E-03	3,17E-03	1,17E-02	3,07E-03	6,73E-03
0,4	7,51E-05	4,60E-03	1,36E-02	1,40E-02	8,09E-03
0,45	6,44E-03	9,77E-03	1,30E-02	1,08E-02	9,99E-03
0,5	6,51E-03	1,00E-02	1,32E-02	2,63E-02	1,40E-02
0,55	1,41E-02	1,59E-02	2,12E-02	4,46E-02	2,40E-02
0,6	1,71E-02	1,19E-02	1,85E-02	4,57E-02	2,33E-02
0,65	8,26E-03	3,12E-02	2,46E-02	4,95E-02	2,84E-02
0,7	1,19E-02	3,58E-02	3,35E-02	7,61E-02	3,93E-02
0,75	3,83E-02	2,27E-02	3,24E-02	8,26E-02	4,40E-02
0,8	2,92E-03	7,95E-02	4,53E-02	1,17E-01	6,12E-02
0,85	2,59E-01	1,65E+00	2,54E-02	1,11E-01	5,12E-01
0,9	1,95E-01	1,18E+01	7,93E-02	1,45E-01	3,06E+00

Cuadro 3.1: Error numérico del índice de Whittle en cada estado con respecto a su valor teórico para cada valor de γ en el problema de **dinámica circular**.

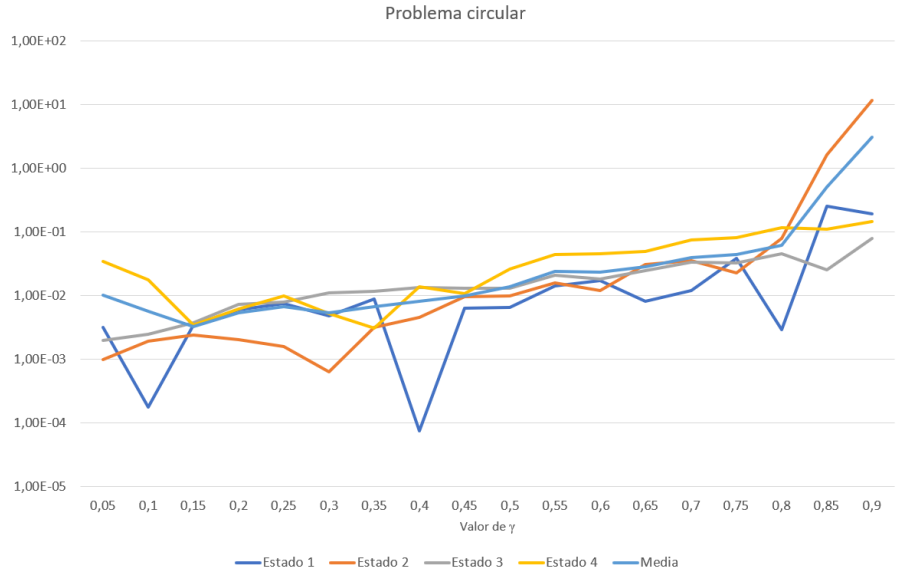


Figura 3.6: Error numérico de los índices de Whittle para cada estado en función del parámetro γ para el problema de **dinámica circular**.

Al utilizar $\gamma = 0{,}3$ obtenemos, a partir de la ecuación (3.29), los índices teóricos $g(1) = -0{,}15, g(2) = 0{,}15, g(3) = 0{,}2013, g(4) = -0{,}2013$. En la gráfica 3.7 vemos como los valores numéricos de los índices convergen asintóticamente a los valores teóricos.

Por otro lado, en la gráfica 3.8 comparamos la recompensa de las primeras 4000 iteraciones durante el entrenamiento del algoritmo frente a las recompensas empleando los valores teóricos de los índices

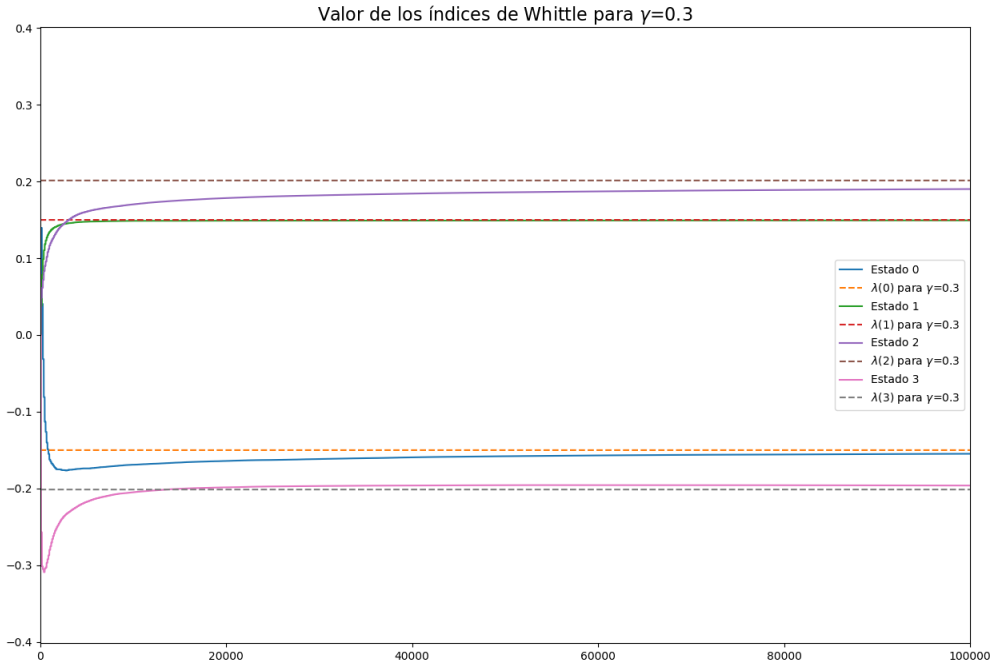


Figura 3.7: Índices de Whittle para el **problema con dinámica circular** para $\gamma = 0{,}3$

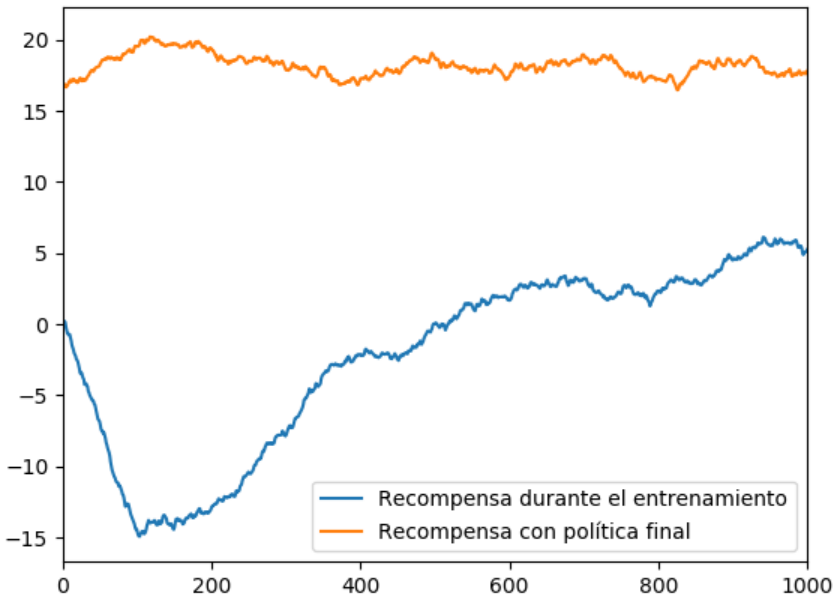


Figura 3.8: **Problema con dinámica circular**: Comparación entre las recompensas durante el entrenamiento, empleando $\varepsilon = 0{,}1$, frente a la recompensa obtenida con la política definida desde el principio.

desde el principio. Una vez se ha definido la política en el entrenamiento, la diferencia de rendimiento frente a emplear la recompensa óptima desde el principio proviene del uso del algoritmo "Epsilon-

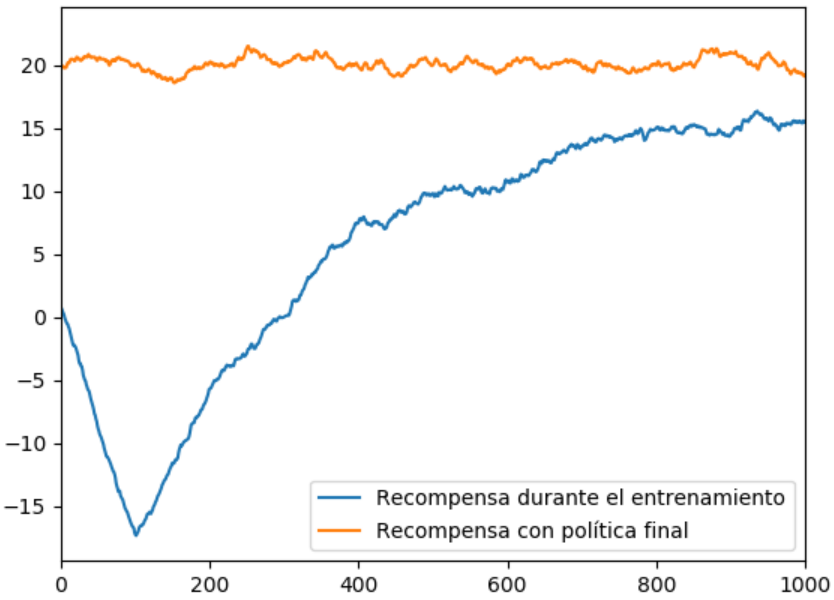


Figura 3.9: **Problema con dinámica circular**: Comparación entre las recompensas durante el entrenamiento, empleando $\varepsilon = 0{,}01$, frente a la recompensa obtenida con la política definida desde el principio.

greedy": En la gráfica 3.8 empleamos un valor de $\varepsilon = 0{,}1$, de modo que un 10 % de las veces tomamos una acción que no es necesariamente óptima. En la gráfica 3.9 tomamos $\varepsilon = 0{,}01$, de modo que solo un 1 % de las veces realizamos exploración frente a explotación de la información. En este caso, no solo el algoritmo define antes su política, sino que la recompensa que obtiene es mucho más cercana a la recompensa óptima.

3.3.2. Problema con reinicio

En el problema del *reinicio*, empleamos 5 estados distintos, donde la recompensa sí depende de la acción que tomamos en cada estado: si la acción es positiva, volvemos al primer estado de la cadena con probabilidad 1, pero la recompensa es 0, mientras que una acción pasiva implica avanzar con probabilidad $9/10$ (a menos de que estemos en el último estado de la cadena, en cuyo caso nos mantenemos en ese estado) o volvemos al primer estado con probabilidad $1/10$. La recompensa de cada estado con acción pasiva es $R_0(k) = 0{,}9^k$, donde k es el número del estado. En la sección 3.1.2 vimos como los valores teóricos de los índices de Whittle para este problema eran:

$$g(1) = -\frac{9}{10} \tag{3.30a}$$

$$g(2) = \frac{81\gamma}{1000} - \frac{81}{100} \tag{3.30b}$$

$$g(3) = \frac{6561\gamma^2}{100000} + \frac{1539\gamma}{10000} - \frac{729}{1000} \tag{3.30c}$$

$$g(4) = \frac{81\left(6561\gamma^3 + 15390\gamma^2 + 27100\gamma - 81000\right)}{10000000} \tag{3.30d}$$

$$g(5) = \frac{43046721\gamma^4}{1000000000} + \frac{10097379\gamma^3}{100000000} + \frac{1778031\gamma^2}{10000000} + \frac{278559\gamma}{1000000} - \frac{59049}{100000} \tag{3.30e}$$

En la tabla 3.2 y la gráfica 3.10 representamos el error numérico del cálculo del índice de Whittle tras un entrenamiento de 100000 iteraciones, para distintos valores de γ. Al igual que en el caso de dinámica circular, cuanto mayor es el valor del parámetro de descuento, mayor es el error numérico, llegándose a disparar a partir de $\gamma = 0{,}75$. Debido al rápido crecimiento de los errores numéricos con el parámetro de descuento, emplearemos en el análisis de este caso $\gamma = 0{,}1$, el cual, si bien relega muy poco peso en las recompensas de los pasos futuros, garantiza un cálculo óptimo de los índices de Whittle.

γ	Estado 1	Estado 2	Estado 3	Estado 4	Estado 5	Promedio
0,05	1,22E-02	1,21E-02	1,21E-02	1,46E-02	2,82E-02	1,58E-02
0,1	1,43E-02	1,29E-02	1,12E-02	1,45E-02	1,54E-02	1,37E-02
0,15	1,59E-02	1,48E-02	1,56E-02	1,70E-02	5,17E-02	2,30E-02
0,2	1,75E-02	1,94E-02	2,65E-02	2,92E-02	1,40E-01	4,64E-02
0,25	1,92E-02	2,28E-02	2,86E-02	4,96E-02	1,45E-01	5,31E-02
0,3	2,08E-02	2,21E-02	2,29E-02	4,74E-02	4,74E-02	3,21E-02
0,35	2,05E-02	2,27E-02	3,83E-02	8,79E-02	1,81E-01	7,02E-02
0,4	2,42E-02	2,43E-02	1,71E-02	6,92E-02	1,99E-01	6,68E-02
0,45	2,37E-02	2,40E-02	1,09E-02	7,00E-02	2,15E-01	6,87E-02
0,5	2,95E-02	2,35E-02	2,66E-02	2,03E-01	3,08E-01	1,18E-01
0,55	3,16E-02	2,49E-02	2,19E-02	2,66E-01	5,18E-01	1,72E-01
0,6	2,34E-02	1,35E-02	1,17E-01	7,35E-01	5,79E-01	2,94E-01
0,65	2,09E-02	1,01E-02	1,33E-01	1,25E+00	4,52E-01	3,74E-01
0,7	1,26E-02	3,77E-03	5,23E-01	3,31E+00	3,72E-01	8,43E-01
0,75	3,16E-02	2,38E-02	2,33E+00	6,00E+00	1,97E+00	2,07E+00
0,8	2,61E-02	1,31E-02	1,88E+01	1,36E+01	2,48E-01	6,54E+00
0,85	1,50E-02	5,40E-03	8,75E+01	1,96E+02	3,83E+00	5,74E+01
0,9	2,37E-02	8,87E-03	3,49E+03	2,62E+02	1,33E+02	7,76E+02

Cuadro 3.2: Error numérico del índice de Whittle en cada estado con respecto a su valor teórico para cada valor de γ en el problema de **reinicio**.

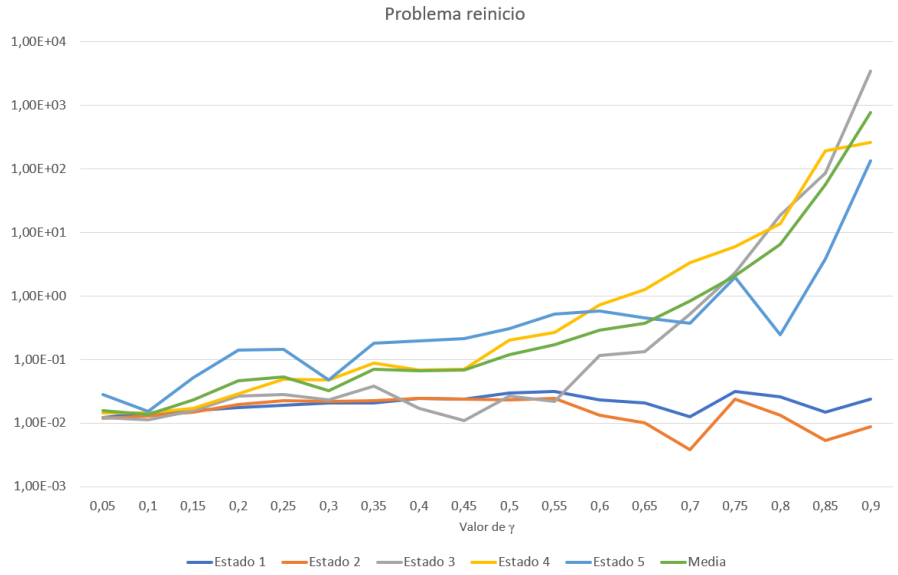


Figura 3.10: Error numérico de los índices de Whittle para cada estado en función del parámetro γ para el problema de reinicio

Con un valor de $\gamma = 0{,}1$, los índices de Whittle en (3.30) pasan a ser $g(1) = -0{,}9, g(2) = -0{,}8019, g(3) = -0{,}713, g(4) = -0{,}6328, g(5) = -0{,}5608$. En la gráfica 3.11, vemos como los índices numéricos de los 5 estados convergen a los valores teóricos de los índices de Whittle. Este problema tiene especial interés debido a que, dada su dinámica, los últimos estados son mucho menos visitados que los primeros, y por lo tanto, reciben menos visitas durante el entrenamiento.

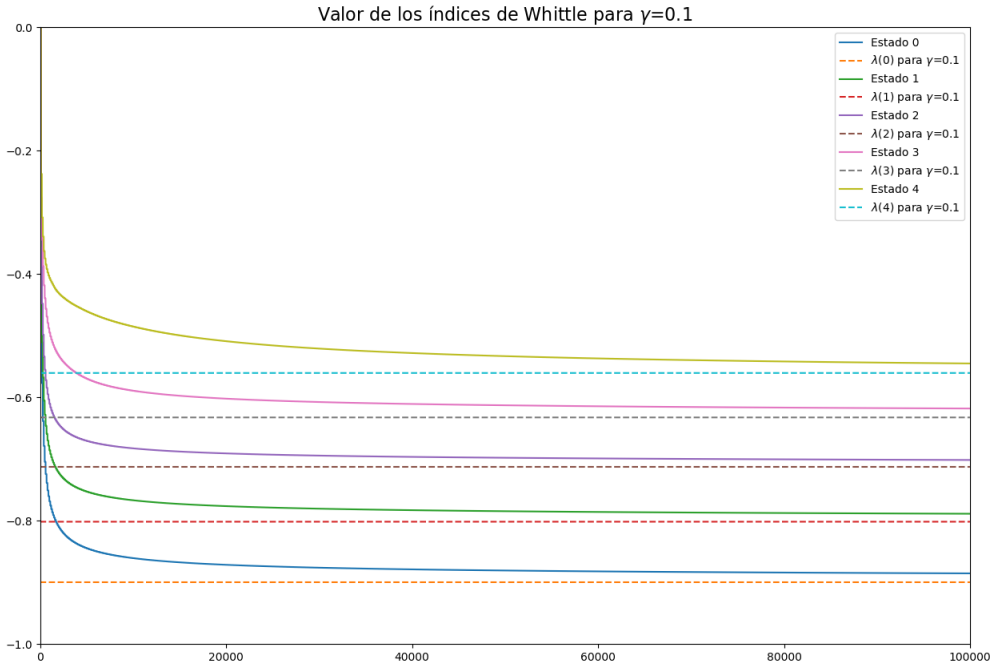


Figura 3.11: Índices de Whittle para el **problema con reinicio** para $\gamma = 0{,}1$

En la gráfica 3.12 comparamos la "running time average reward" obtenida por el algoritmo durante las primeras 900 iteraciones durante el entrenamiento frente al obtenido utilizando una política óptima desde el primer momento. Una vez definida la política durante el entrenamiento, la diferencia en rendimiento entre ambos se debe, al igual que en el problema de dinámica circular, al uso del algoritmo "Epsilon-greedy". Al emplear un valor $\varepsilon = 0{,}01$ en lugar de 0,1 en la gráfica 3.13 reducimos el margen de rendimiento entre el algoritmo durante el entrenamiento y el algoritmo empleando la política óptima desde el principio.

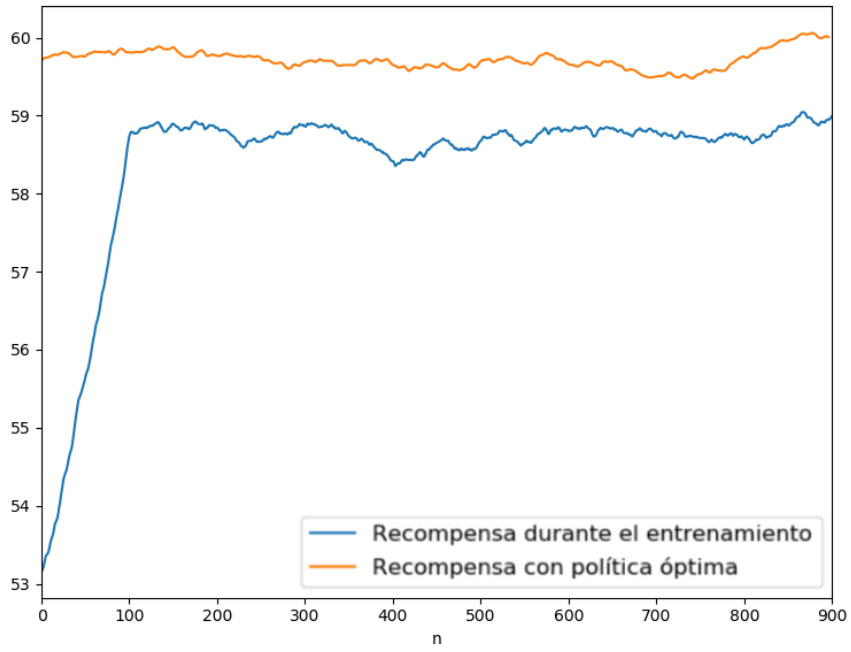


Figura 3.12: **Problema con *restart***: Comparación entre las recompensas durante el entrenamiento, empleando $\varepsilon = 0{,}1$, frente a la recompensa obtenida con la política definida desde el principio.

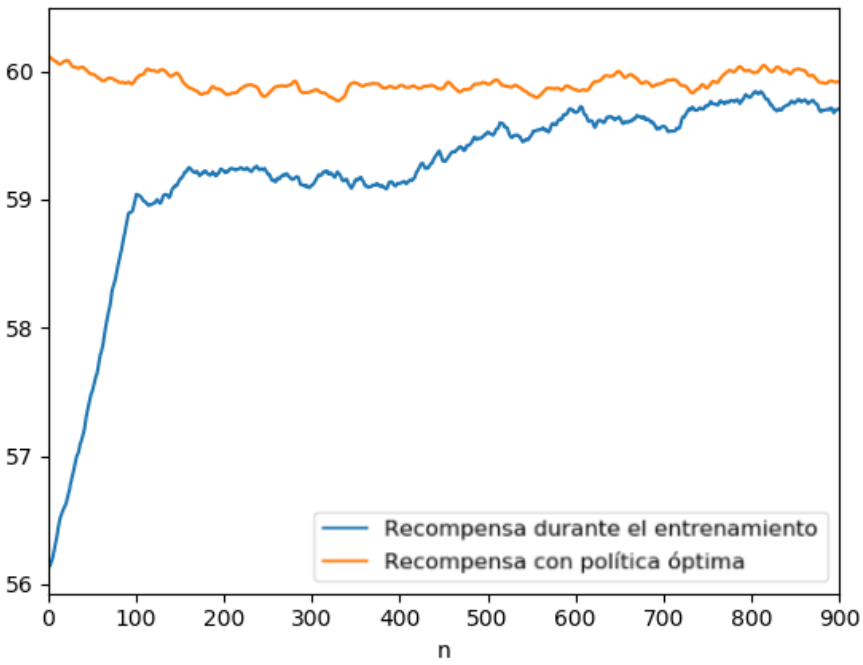


Figura 3.13: **Problema con reinicio**: Comparación entre las recompensas durante el entrenamiento, empleando $\varepsilon = 0{,}01$, frente a la recompensa obtenida con la política definida desde el principio.

Capítulo 4

Conclusiones finales

En los anteriores capítulos, hemos planteado un algoritmo para el cálculo de los índices de Whittle para dos problemas con dinámicas y recompensas distintas: uno con una dinámica "circular" en el que el último estado de la cadena de Markov conecta con el primero y otro problema con "reinicio", en el que una de las acciones transporta instantáneamente al primer estado de la cadena. En ambos casos, hemos obtenido los valores teóricos de los índices de Whittle para cada estado en función del factor de descuento γ para el criterio de "optimalidad" de *recompensa descontada en el horizonte infinito de tiempo* (ecuaciones (3.29) y (3.30)).

Para crear unas condiciones de convergencia favorables para nuestro algoritmo, hemos empleado dos escalas de tiempo en el cálculo de los índices de Whittle: por un lado, una escala "rápida", equivalente a *una actualización por iteración* para el cálculo de los Q-values, para determinar el valor de cada acción en cada estado, y una escala "lenta", equivalente a *una actualización por cada múltiplo del número de bandidos* (ecuación (2.13b)), para el cálculo de los índices de Whittle a partir de los Q-values, y evaluar así cada estado. Debido al uso de los índices de Whittle para el cálculo de los Q-values (2.11), ambos sistemas de ecuaciones se encuentran acoplados. El sistema de dos escalas de tiempo nos permite desacoplar estas ecuaciones y obtener las condiciones de convergencia expuestas en la sección 3.2.1.

En la sección 3.3 hemos llevado a cabo este algoritmo para los casos de *dinámica circular* y el *problema con reinicio*, empleando para cada uno de ellos un factor $\gamma = 0{,}3$ y $0{,}1$ respectivamente. En ambos casos, se ha estudiado la convergencia a los valores teóricos de sus índices de Whittle, calculados en el capítulo 3, con valores $g(1) = -0{,}15, g(2) = 0{,}15, g(3) = 0{,}2013, g(4) = -0{,}2013$ y $g(1) = -0{,}9, g(2) = -0{,}8019, g(3) = -0{,}713, g(4) = -0{,}6328, g(5) = -0{,}5608$ para cada problema. Como se puede observar en las gráficas 3.7 y 3.11, a medida que aumentamos el número de iteraciones, y por lo tanto el tiempo de aprendizaje, nos acercamos más a los valores teóricos óptimos de estos índices.

Por otro lado, las gráficas 3.6 y 3.10 nos muestran un aspecto importante del comportamiento de nuestor algoritmo: la dependencia en la convergencia de los índices en función de γ. En los dos casos que hemos discutido en este trabajo, el error numérico en los índices de Whittle aumenta al acercarnos al valor de $\gamma = 1$. Como podemos ver en la ecuación (2.11), este término regula el peso del Q-value del siguiente estado en el cálculo del Q-value del estado actual. Debido a que los Q-values están fundamentados por las recompensas de los estados (ecuación (2.10)) una explicación de este comportamiento es la acumulación de recompensas futuras a medida que aumentamos el valor de γ, provocando inestabilidades en el cálculo de los índices de Whittle (ecuación (2.12)). En los dos problemas que hemos planteado en este trabajo, hemos empleado dos funciones de recompensas distintos:

- En el problema de **dinámica circular** (sección 3.1.1) empleamos una función de recompensa

$$R(1) = -1, R(2) = 0, R(3) = 0, R(4) = 1$$

  Para los 4 estados que componen la cadena de Markov de ese sistema.

- En el problema de **reinicio** (sección 3.1.2) empleamos una función de recompensa

$$R_0(k) = 0{,}9^k$$
$$R_1(k) = 0$$

  Para los 5 estados de la cadena de Markov.

Si observamos la evolución de los errores numéricos en las gráficas 3.6 y 3.10, vemos como el error en el caso del problema de dinámica circular es consistentemente más pequeño que en el de reinicio, ya que siempre tiene más bandidos con recompensas de valor nulo que en el segundo problema. Esta *menor densidad de recompensas* hace que el valor acumulado de los Q-values sea más pequeño en el problema de dinámica circular, y se pueda emplear valores de γ más altos antes de que aparezcan inestabilidades.

Bibliografía

[1] D. A. Berry and B. Fristedt, "Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability)," *London: Chapman and Hall*, vol. 5, pp. 71–87, 1985.

[2] J. Fu, B. Moran, J. Guo, E. W. Wong, and M. Zukerman, "Asymptotically optimal job assignment for energy-efficient processor-sharing server farms," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 4008–4023, 2016.

[3] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5547–5567, 2010.

[4] S. Deo, S. Iravani, T. Jiang, K. Smilowitz, and S. Samuelson, "Improving health outcomes through better capacity allocation in a community-based chronic care model," *Operations Research*, vol. 61, no. 6, pp. 1277–1294, 2013.

[5] A. V. den Boer, "Dynamic pricing and learning: historical origins, current research, and new directions," *Surveys in operations research and management science*, vol. 20, no. 1, pp. 1–18, 2015.

[6] P. Whittle, "Restless bandits: Activity allocation in a changing world," *Journal of applied probability*, vol. 25, no. A, pp. 287–298, 1988.

[7] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[8] B. Widrow and M. E. Hoff, "Adaptive switching circuits," tech. rep., Stanford Univ Ca Stanford Electronics Labs, 1960.

[9] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[10] R. J. Solomonoff, "A formal theory of inductive inference. part ii," *Information and control*, vol. 7, no. 2, pp. 224–254, 1964.

[11] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] t. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[15] O. Hegazy, O. S. Soliman, and M. A. Salam, "A machine learning model for stock market prediction," *arXiv preprint arXiv:1402.7351*, 2014.

[16] D. Ciregan, U. Meier, and t. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3642–3649, IEEE, 2012.

[17] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[18] M. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, p. 3, 2012.

[19] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena scientific Belmont, MA, 1995.

[20] R. E. Bellman and S. E. Dreyfus, *Applied dynamic programming*. Princeton university press, 2015.

[21] L. Mitten, "An analytic solution to the least cost testing sequence problem," *Journal of Industrial Engineering*, vol. 11, no. 1, p. 17, 1960.

[22] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 41, no. 2, pp. 148–164, 1979.

[23] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[24] F. S. Melo, "Convergence of q-learning: A simple proof," *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

[25] D. Ruiz-Hernandez, *Indexable restless bandits: Index policies for some families of stochastic scheduling and dynamic allocation problems*. VDM Publishing, 2008.

[26] R. Bauernschmitt and R. Ahlrichs, "Treatment of electronic excitations within the adiabatic approximation of time dependent density functional theory," *Chemical Physics Letters*, vol. 256, no. 4-5, pp. 454–464, 1996.

[27] N. C. Handy and A. M. Lee, "The adiabatic approximation," *Chemical physics letters*, vol. 252, no. 5-6, pp. 425–430, 1996.

[28] M. Sarandy and D. Lidar, "Adiabatic approximation in open quantum systems," *Physical Review A*, vol. 71, no. 1, p. 012331, 2005.

[29] M. Born and J. R. Oppenheimer, "On the quantum theory of molecules," *Collection of articles to the multimedia electronic educational and methodical complex in the discipline physics of the atom and atomic phenomena ed. by Shundalov MB; BSU, Faculty of Physics.*, 1927.

[30] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[31] K. Avrachenkov and V. S. Borkar, "Whittle index based q-learning for restless bandits with average reward," *arXiv preprint arXiv:2004.14427*, 2020.

[32] J. Fu, Y. Nazarathy, S. Moka, and P. G. Taylor, "Towards q-learning the whittle index for restless bandits," in *2019 Australian & New Zealand Control Conference (ANZCC)*, pp. 249–254, IEEE, 2019.

[33] P. Jacko, "Dynamic priority allocation in restless bandit models," 2010.

[34] C. Lakshminarayanan and S. Bhatnagar, "A stability criterion for two timescale stochastic approximation schemes," *Automatica*, vol. 79, pp. 108–114, 2017.

[35] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*, vol. 48. Springer, 2009.

[36] T. Jaakkola, M. I. Jordan, and S. P. Singh, "Convergence of stochastic iterative dynamic programming algorithms," in *Advances in neural information processing systems*, pp. 703–710, 1994.

[37] J. Abounadi, D. Bertsekas, and V. S. Borkar, "Learning algorithms for markov decision processes with average cost," *SIAM Journal on Control and Optimization*, vol. 40, no. 3, pp. 681–698, 2001.

[38] S. J. Russell and P. Norvig, "Artificial intelligence-a modern approach, third international edition.," 2010.

[39] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Machine Learning Proceedings 1995*, pp. 30–37, Elsevier, 1995.