Procesos de decisión de Markov y Q-learning

Universidad Autónoma de Madrid

Facultad de Ciencias

Departamento de Matemáticas

Trabajo de fin de máster

Máster en Matemáticas y Aplicaciones

Autor:
Santiago Babío Fernández

Beneficiario de la Ayuda para el Fomento de la Investigación en Estudios de Máster

Director:
Enrique Zuazua Iriondo

Fecha:
Julio de 2021

Curso 2020-2021

Resumen

El problema abstracto de un agente cuyo estado evoluciona no solo influido por factores externos y/o aleatorios sino también de acuerdo a decisiones o iniciativas tomadas por éste es una coyuntura ubicua en las aplicaciones de las matemáticas, desde la ingeniería hasta las ciencias sociales. Suponiendo que nos es dado un cierto modelo sobre la dinámica subyacente y la manera en la que ésta es afectada por las acciones ejercidas, el conjunto de técnicas matemáticas que analiza las prácticas más adecuadas a la hora de determinar las actuaciones recibe el nombre de Teoría de Control.

En muchas ocasiones, no obstante, nos encontramos con situaciones en las que tenemos la capacidad de controlar un sistema para el cuál desconocemos las leyes fundamentales de su evolución. En su lugar, solo disponemos de la posibilidad de realizar observaciones de su comportamiento mediante bases de datos o simulaciones y ensayos. Ante la falta de un modelo subyacente, no es posible plantear directamente el problema en el marco de trabajo de la Teoría de Control; sería necesario emplear previamente alguna técnica de modelado y/o de problemas inversos para identificar qué objeto matemático puede describir la dinámica que deseamos controlar.

La recurrencia de estas situaciones sugirió la conveniencia de encontrar un procedimiento para tratar estos problemas de forma sistemática y unificada. Entre las múltiples propuestas destacó la metodología de Q-learning, cuya novedad fundamental fue la de abordar la tarea en un solo paso. En lugar de intentar entender primero el entorno en el se encuentra un agente para después decidir qué acciones tomar, el algoritmo de Q-learning trata de *aprender* directamente a actuar en un mundo que desconoce. La manera en la que esta tarea se lleva a cabo parte de, en primer lugar, suponer - ante la ausencia de un modelo más concreto sobre la dinámica subyacente - que la evolución está gobernada por un proceso suficientemente general (en concreto, un proceso de decisión de Markov). A partir de ahí, se analiza el caso general de estas dinámicas y se manipula inteligentemente para estudiar cómo aproximar directamente la actuación óptima sin necesidad de conocer la gran cantidad de parámetros que determinarían el modelo en su totalidad.

El objetivo de este trabajo es desarrollar las ideas fundamentales y la estructura matemática de esta técnica, que hoy en día supone una piedra angular de lo que se conoce como *reinforcement learning*. Con este fin, la exposición se divide en dos grandes bloques. En primer lugar, se estudian los resultados principales de los ya mencionados Procesos de Decisión de Markov, un modelo probabilístico habitual para tratar la evolución de un agente cuyas acciones influyen en sus estados futuros. En segundo lugar, se introduce el algoritmo de Q-learning, que combina ideas de la Teoría de Control de Procesos de Decisión de Markov junto con razonamientos estadísticos para intentar estimar cuál es el conjunto de decisiones óptimas a partir de la mera observación de ensayos.

Abstract

The abstract problem of an agent whose state evolves according to not only exogenous and random factors but also the actions and decisions it takes is a ubiquitous problem in a wide range of fields, from engineering to social sciences. If a certain model is given describing how the underlying dynamics are affected by the executed actions, the set of mathematical techniques that analyse the procedures to determine how to best manoeuvre are referred to as Control Theory.

However, it is not unusual to encounter situations in which we are capable of controlling a system whose fundamental laws of evolution are unknown. All that is available instead is the possibility to observe its behaviour via either simulations or accumulated data. Providing no explicit mathematical model is at our disposal, it is not possible to directly employ Control Theory's framework; it would be necessary to previously apply some modelling and/or inverse problem technique to identify which mathematical object can describe the dynamics we wish to control.

The recurrence of these circumstances encouraged researchers to suggest a procedure to deal with these problems in a unified, systematic manner. Among the many different proposals, the Q-learning methodology stood out: its novelty was tackling the task in one single step. Instead of trying to firstly understand the agent's environment and then deciding what actions to take, the Q-learning algorithm aims directly at *learning* to perform in a world it does not understand. To this end, it first assumes - in the absence of a better guess - that the evolution is governed by a sufficiently general stochastic process (more precisely, a Markov Decision Process). From then on, these dynamics are studied to find a way to approximate the optimal policy via simulations even if the large amount of parameters that determine the model are not fully known.

The aim of this work is to elaborate on the fundamental ideas and the mathematical structure of this technique, a cornerstone of the so-called *reinforcement learning*. To this end, the exposition is divided into two parts. Firstly, the main results of the above-mentioned Markov Decision Processes (MDPs) are discussed. MDPs are a probabilistic model often used to study the evolution of an agent whose actions influence the upcoming states. Secondly, the Q-learning algorithm is analysed: by combining MDPs' Control Theory ideas and statistical arguments, it will provide a way to estimate the optimal policy merely based on simulations.

Índice general

CAPÍTULO 1
Introducción

A modo de ilustración y como ejemplo del tipo de situaciones que perseguimos analizar, consideremos el ejemplo de un individuo participando en un juego de mesa: dada una cierta situación de la partida, el instante posterior puede venir determinado por una mezcla de restricciones (cuestiones estructurales del sistema), acciones elegidas por el propio jugador, decisiones de los contrincantes y azar. De entre las múltiples posibilidades que se pueden presentar a la hora de decidir, nuestro protagonista trata de escoger sus movimientos de modo que maximice un determinado criterio para un horizonte dado. Adecuadamente abstraídas, estas situaciones son altamente generales y se pueden observar en muy distintos marcos de trabajo; por ejemplo, el vuelo de un dron entre dos puntos del espacio - que obedece a las leyes de la mecánica clásica, controles ejecutados por el autopiloto y perturbaciones aleatorias (como un ráfaga de viento) - puede ser abordado con un enfoque relativamente parecido.

La rama de las matemáticas que se pregunta sobre, en primer lugar, si es posible alcanzar unos ciertos objetivos para las posibilidades de maniobra de las que se dispone, y, en segundo lugar, cómo hacerlo de manera *óptima* recibe el nombre de Teoría de Control. Aunque existen muchas posibles variantes, un ejemplo del tipo de problemas que abordaba en un principio es lo que se conoce como el problema de Mayer. Dado un intervalo temporal $[0,T]$, una ecuación diferencial $y' = f(t,y,u)$, un punto de partida $x \in \mathbb{R}^n$ y un criterio para valorar la bondad de la posición final (que puede venir encapsulado por una función $g : \mathbb{R}^n \to \mathbb{R}$), el objetivo es escoger una función $u : [0,T] \to \mathbb{R}^m$ para buscar una trayectoria $y_{u,x}(t)$ tal que maximice $g(y_{u,x}(T))$ y satisfaga

$$\begin{cases} y'_{u,x}(t) = f(t,y,u), & t \in [0,T] \\ y_{u,x}(0) = x. \end{cases}$$

Con el desarrollo en el siglo XX de la teoría de probabilidad se comenzaron a considerar también problemas de control para sistemas en cuya evolución intervenían fenómenos probabilistas, sobre los cuales se va a centrar este trabajo. Un ejemplo sencillo sobre el que nos apoyamos para fijar ideas es el del *paseo aleatorio* sobre $\mathbb{Z}$ controlado. Consideremos un agente con posición inicial $k \in \mathbb{Z}$ cuyo objetivo es

maximizar, dado un número finito de instantes temporales N y una función $g : \mathbb{Z} \to \mathbb{R}$, el valor de $g(X_N)$. La dinámica a la que está sometido es la siguiente: en cualquier época temporal $n = 0, 1, ..., N-1$, se ha de elegir de entre el conjunto de acciones $A = \{-1, 0, +1\}$ si se prefiere ir a la derecha ($C_n = +1$), izquierda ($C_n = -1$) o quedarse quieto ($C_n = 0$), y en base a esa actuación tiene lugar una transición estocástica con distintas distribuciones.

- $\mathbb{P}(X_{n+1} = j | X_n = k, C_n = +1) = \frac{1}{2}$ si $j = k+2$, $j = k$;
- $\mathbb{P}(X_{n+1} = j | X_n = k, C_n = -1) = \frac{1}{2}$ si $j = k-2$, $j = k$;
- $\mathbb{P}(X_{n+1} = j | X_n = k, C_n = 0) = \frac{1}{2}$ si $j = k-1$, $j = k+1$.

Es evidente que la situación actual difiere del problema anterior en el sentido de que, dada una ley de control, la posición final ya no nos viene totalmente determinada como en el ejemplo previo. En su lugar, lo que vamos a obtener ahora es que a cada conjunto de acciones que se decida tomar le va a corresponder una distribución de la variable aleatoria X_N; además, puesto que $g(X_N)$ puede tomar distintos valores para una misma sucesión de decisiones, sobre lo que vamos a trabajar ahora son las ganancias en esperanza.

Este ejemplo es una sencilla particularidad de un conjunto de problemas de *decisión multi-etapa* que hoy en día se conocen como procesos de decisión de Markov (*Markov decission processes*, MDP). Tanto las ideas subyacentes como las herramientas y métodos utilizados en esta teoría tienen la huella imborrable del matemático aplicado R.E. Bellman, responsable principal de su desarrollo a finales de los años 40 y entrados los 50 ([14]). Entre otras cosas, a Bellman se le atribuye el *principio de optimalidad*, que se puede expresar en palabras del siguiente modo:

Principio de optimalidad. *Una sucesión de decisiones óptima satisface que, dados un estado y un instante inicial cualquiera, las acciones de los instantes posteriores constituyen una manera de actuar óptima para el estado resultante tras la primera decisión.*

A partir de un enunciado tan intuitivo y sencillo como éste Bellman desarrolló unas técnicas que no solo permitieron abordar con éxito el estudio de los procesos de decisión de Markov, sino que revolucionaron el enfoque utilizado en la teoría clásica de control óptimo de sistemas deterministas como el presentado al comienzo de este capítulo. Hoy en día el *principio de optimalidad* vertebra toda la teoría de control óptimo, y aunque en esta memoria solo nos vamos a centrar en problemas del tipo de decisión de Markov cabe destacar que muchas de las ideas se aplican de manera totalmente análoga tanto en el caso de tiempo continuo determinista (ver [17], [7]) como estocástico ([20]).

De este modo, como veremos en la primera parte de este trabajo para el caso estocástico a tiempo y espacio discreto, guiados por el trabajo de Bellman se podrá concluir que en muchos casos de interés el problema de control está resuelto: se dispone de algoritmos para calcular leyes de control arbitrariamente *cercanas* a la manera óptima de actuar. Para situaciones que además se ajustan a planteamientos relativamente habituales y en las que se dispone de modelos precisos, la aplicación de la teoría de control se reduce a implementaciones de *recetas* ya ampliamente establecidas y aceptadas. Este es el caso cuando se trabaja con sistemas cuya dinámica está profundamente estudiada y entendida, como ocurre con - por ejemplo - cuerpos regidos por la mecánica clásica.

Sin embargo, no es extraño toparse con la necesidad o el interés de controlar sistemas cuya dinámica subyacente no se comprende en profundidad. Como veremos, la generalidad de los procesos de decisión de Markov invita a considerar que quizá puedan ser una buena herramienta para estructurar matemáticamente una gran cantidad de problemas de control observados en la vida real. Así pues, una vez se han estudiado este tipo de modelos, la primera idea que se le puede ocurrir al matemático aplicado a la hora de intentar controlar un sistema desconocido parcial o totalmente es proceder en dos pasos:

1. Utilizando observaciones (ya sea a través de datos almacenados o mediante simulaciones/ensayos), realizar una estimación de los parámetros del proceso de decisión de Markov. A modo de ilustración, este proceso sería muy similar - pero aún más costoso - a estimar las matrices de transición de una cadena de Markov.

2. Con esa aproximación que se ha obtenido para el proceso de decisión de Markov subyacente, se procede a utilizar técnicas basadas en el *principio de optimalidad* para encontrar una ley de control óptima para esa estimación concreta.

Aunque parece sólido en una primera aproximación, rápidamente surgen dos preguntas respecto a la efectividad y eficiencia de dicho proceder. En primer lugar, es natural preguntarse si los dos pasos se acoplan adecuadamente. Si tenemos un buen método de estimación, sabemos que nuestro modelo aproximado convergerá al *verdadero*; por otro lado, dado un modelo aproximado, veremos cómo encontrar la ley de control óptima. Sin embargo, ¿convergerá la sucesión de políticas de control óptimas para modelos aproximados a la política de control óptima para el modelo *verdadero*? Este interrogante está ya documentado en el campo de la ingeniería de control, y en algunas circunstancias ha dado lugar a una cantidad significativa de esfuerzo - de donde han surgido técnicas útiles en la práctica como el Model Predictive Control (ver [6]). Desde el punto de vista teórico, no obstante, es claro que el tratamiento riguroso de esta interacción y la obtención de garantías en el límite no son en absoluto triviales.

Por otro lado, se puede adelantar que el paso 1 de los recién mencionados es un proceso computacional muy costoso cuya finalidad es obtener unos valores que solo nos

interesan de manera indirecta. Dado que en algunas circunstancias solamente estamos interesados en conocer cuál es la política de control óptima del modelo *verdadero*, resulta lógico preguntarse si es posible encontrar una forma de aproximar esa política óptima de manera directa, ahorrando cálculos intermedios. Algoritmos de este tipo no solo nos evitarían la cuestión teórica planteada en el párrafo anterior, sino que también es de esperar que en algunas de sus versiones y bajo ciertas circunstancias se comporten de manera más eficiente.

Además, a diferencia de la aplicación sucesiva de los pasos 1 y 2, la mejoría de las políticas estimadas con métodos de esta naturaleza sería progresiva en vez de por bloques, por lo que los aumentos de precisión debidos a la incorporación de nuevas observaciones se producirían en tiempo real. Entre otras circunstancias, esta sería una característica interesante para situaciones en las que la política se estima a la vez que se está operando en el entorno. También dotaría a los métodos de mayor flexibilidad, pues ya no cabría preguntarse cómo distribuir el trabajo entre la construcción del modelo y la optimización posterior: la manera de proceder sería la misma independientemente de la cantidad de esfuerzo computacional de la que se disponga.

De entre la familia de algoritmos que aspiran a aproximar de manera directa esa política, una de las técnicas más influyentes en la literatura moderna es la que se conoce como Q-learning, introducida en 1989 por C.J.C.H. Watkins ([41]). La segunda parte de este trabajo se centrará en demostrar que el procedimiento que ésta sugiere efectivamente converge bajo condiciones muy generales, y dará estimaciones sobre la velocidad y el carácter de la convergencia para algunos casos concretos.

Con el objeto de poner en el contexto el impacto de esta técnica, cabe destacar que el método de Q-learning es uno de los enfoques fundamentales de lo que se conoce como *reinforcement learning* ([27]) - que junto con el *supervised learning* y el *unsupervised learning* constituyen los tres paradigmas básicos del *machine learning*. Respecto al *aprendizaje por refuerzo*, D.P. Bertsekas y J.N. Tsitsiklis comentaban en 1996 lo siguiente ([4]):

> *"A few years ago our curiosity was aroused by reports on new methods in reinforcement learning, a field that was developed primarily within the artificial intelligence community [...]. Our first impression was that the new methods were ambitious, overly optimistic, and lacked firm foundation [...]. [...] we believe our initial impressions were largely correct [...], but for reasons that we know understand much better, it [reinforcement learning] does have the potential of success with important and challenging problems. [...] Furthermore, the methodology has [now] a logical structure and a mathematical foundation [...]."*

Los éxitos y disrupciones que en un principio atrajeron la curiosidad de estos dos matemáticos aplicados se han continuado sucediendo en los años recientes; entre las demostraciones tecnológicas más sorprendentes, se encuentran las llevadas a cabo a largo de la década de 2010 por la empresa DeepMind Technologies (adquirida por Google en 2014). Utilizando diferentes técnicas entre las que se encontraban extensiones de Q-learning, este grupo de investigadores desarrolló métodos para entrenar a un agente y obtener resultados similares a los de un humano en juegos *arcade* de la videoconsola Atari 2600 (ver [30]). Lo interesante del enfoque empleado en esos ejemplos no es solo que la complejidad de los juegos obliga a evitar el uso de fuerza bruta, sino también el hecho de que el algoritmo utilizado en los distintos casos es el mismo y no requiere la introducción de técnicas *ad-hoc*. Basándose en ideas y principios no muy diferentes, científicos de esa misma empresa consiguieron también desarrollar dos métodos que permitieron ganar al campeón del mundo en el juego de mesa Go, cuya tratabilidad computacional siempre había supuesto un reto inabordable para la inteligencia artificial previa (ver [32], [33]). Sobre el segundo de los programas informáticos presentados, AlphaGo Zero, destaca el hecho de que alcanzó un nivel sobrehumano en el juego de Go sin la introducción de ningún tipo de información previa.

Figura 1.1: Juegos *arcade* de la videoconsola Atari 2600. En [30] se muestran los resultados que el mismo algoritmo - entrenado utilizando solamente la información de los píxels de la pantalla - obtuvo para distintos juegos. Los resultados son comparables a los obtenidos por un humano.

Aprovechamos esta introducción para destacar que en la literatura (ver, por ejemplo, [41]) no es extraño encontrar interpretaciones más o menos antropomórficas y orgánicas del tipo de técnicas que vamos a presentar en este trabajo. Nosotros preferimos mantenernos al margen de este tipo de debates y nos reducimos a describir la estructura matemática subyacente, que no es más que dar un modelo probabilista y realizar una estimación estadística de la política óptima subyacente.

Destacamos también que en este trabajo nos limitaremos a considerar el caso de tiempo y espacio discreto. Los motivos que fundamentan esta decisión son múltiples; entre ellos, destaca que el algoritmo de Q-learning solo se haya estudiado en profundidad y demostrado su convergencia para este caso. No obstante, debido al enfoque eminentemente computacional del método esta *limitación* no supone en la práctica una restricción real. Por otro lado, la asunción de tiempo y espacio discreto nos brindará un extra de flexibilidad de cara a las aplicaciones del método, en las que será necesario que a la dinámica le subyazga un proceso de decisión de Markov. Como es habitual, siempre será posible encontrar una aproximación discreta razonable de un modelo continuo, mientras que la relación inversa puede no tener sentido.

A modo de ejemplo de cómo modelos discretos aproximan entornos continuos, retomamos el ejemplo del paseo aleatorio *controlado* en $\mathbb{Z}$. Suponiendo que se escoge una ley de control fija $C = c_0 \in \{-1, 0, 1\}$, no es difícil ver que nuestro planteamiento discreto es capaz de replicar adecuadamente el comportamiento de un sistema en $\mathbb{R}$ gobernado por

$$dX_t = c_0 dt + dW_t$$

cuya función de densidad $u_{c_0}(x, t)$ evolucionaría de acuerdo a

$$\frac{\partial u_{c_0}}{\partial t} = \frac{\partial^2 u_{c_0}}{\partial x^2} - c_0 \frac{\partial u_{c_0}}{\partial x}.$$

Inspeccionando las múltiples discretizaciones posibles para ciertas razones entre Δt y Δx, se comprueba que el problema antes propuesto supone una réplica fiel de este mismo fenómeno.

Por otro lado, la consideración de tiempo y espacio continuo introduce una gran cantidad de fenómenos que, si bien interesantes desde el punto de vista técnico, pueden distraer de las ideas principales que se busca transmitir. La riqueza y novedad conceptual del tipo de ideas que introduciremos, el enfoque computacional de las técnicas que detallaremos y la amplia familia de problemas que se busca abordar (desde juegos de mesa hasta gestiones de inventorios) recomiendan mantener la discusión centrada en tiempos y espacios discretos.

Respecto a la línea lógica que se va a seguir, comentar que la idea es centrarse en el problema de la *recompensa total descontada esperada* con un horizonte de tiempo infinito, y utilizar ese problema para ilustrar el modelo probabilista que se utiliza, los resultados teóricos al respecto y la eficiencia y eficacia de la estimación usando Q-learning. En teoría de control, es habitual encontrarse con distintas situaciones que, a pesar de poder abordarse todas con el mismo enfoque, presentan particularidades que requieren la modificación parcial de ciertos razonamientos: el problema de posición final en tiempo finito, el problema del camino más corto, el problema de ganancia media,... Exponer uno de ellos sirve para ilustrar las ideas que vertebran toda esta rama, y, de escoger uno solo, el problema de la *recompensa total descontada esperada* es el más adecuado por su tratabilidad analítica, su ubicuidad y su rol de apoyo en el

desarrollo de otras partes de la teoría.

Concluimos esta introducción describiendo brevemente la estructura de esta memoria. Comenzaremos la exposición con un estudio en el Capítulo 2 de los procesos de decisión de Markov: presentaremos con detalle el modelo probabilístico sobre el que se fundamentan, discutiremos qué métricas utilizar para evaluar la calidad de una cierta sucesión de decisiones y detallaremos un método para encontrar maneras de actuar arbitrariamente cerca del supremo. A modo de ilustración y como ejemplos prácticos, se presentarán también los resultados que se obtienen para dos problemas sencillos. A continuación, en el Capítulo 3 abordaremos el algoritmo de Q-learning. Introduciremos la motivación del método apoyándonos en resultados y conceptos del capítulo anterior, demostraremos su convergencia en el límite y estudiaremos el comportamiento del algoritmo en tiempo finito. Con el fin de verificar la validez de los resultados que arroja, terminaremos el bloque aplicando el algoritmo sobre los problemas que ya habíamos abordado previamente. Por último, en el Capítulo 4, discutiremos las conclusiones y perspectivas de investigación futura que arroja este trabajo, enfatizando la gran cualidad de las técnicas aquí presentadas: su flexibilidad y generalidad.

CAPÍTULO 2
Procesos de decisión de Markov

A lo largo de este capítulo procedemos a exponer la teoría de los procesos de decisión de Markov para el problema de *recompensa total descontada esperada* (*expected total discounted reward*). La cronología de pasos a seguir en los próximas secciones es el procedimiento estándar a la hora de tratar un problema de control óptimo: en primer lugar se formula el modelo, a continuación se estable cuál es el criterio a maximizar/minimizar y finalmente se termina dando un método para encontrar una ley de control arbitrariamente *cercana* al conjunto de acciones óptimo. Incluimos también al final del capítulo dos ejemplos numéricos a modo de aplicación de los resultados teóricos.

2.1. Formulación del modelo

Tal y como se comentó en el Capítulo 1, nuestro objetivo es capturar matemáticamente la dinámica de un agente cuyas acciones influencian la evolución estocástica en tiempo y espacio discreto a la que está sometido. En base a las decisiones que toma, el agente recibe una recompensa en cada instante de tiempo, que puede depender de su estado actual, la acción que ha ejecutado y el estado al que evoluciona. En la variante que consideramos aquí, el número de instantes temporales es infinito y el proceso teóricamente no acaba nunca; no obstante, como se podrá deducir, es muy sencillo incluir el efecto de un mecanismo de parada mediante la introducción de un estado absorbente.

La idea sobre la que vamos a definir el modelo es la de las cadenas de Markov. De manera muy resumida, la esencia va a ser considerar un espacio medible en el que cada elemento ω es una realización; sobre éste, la probabilidad va a venir definida por una matrices de transición en base a la sucesión de decisiones que se escoja. A cada sucesión de acciones le va a corresponder, pues, una probabilidad sobre el espacio de realizaciones.

2.1.1. Elementos del modelo

Una de los primeros ingredientes que hemos de especificar es un conjunto de índices T que encapsula las épocas en las que el agente ejerce las acciones. Como ya se ha dicho, aunque existen modelos para $T \subset \mathbb{R}_+$ o $T \subsetneq \mathbb{N} \cup \{0\}$ (ya sea finito o infinito), aquí nos centraremos en el caso $T = \mathbb{N} \cup \{0\} = \mathbb{N}_0$.

Establecido esto, lo siguiente es determinar el conjunto que utilizamos para caracterizar los posibles estados del sistema. A este espacio de estados lo denotaremos por S, y vamos a imponer que tenga una cantidad numerable de elementos a los que denotaremos mediante letras minúsculas (s,j,...). En ocasiones, nos referiremos a S como espacio medible; en ese caso ha de entenderse que hablamos de $(S, \mathcal{P}(S))$. En cada estado del sistema se podrán realizar una cantidad (a lo sumo numerable) de *acciones* que vendrán recogidas en los conjuntos A_s; a partir de éstos se genera, por comodidad, el conjunto de acciones posibles $A = \cup_{s \in S} A_s$. Una vez más, dado que A tiene una cantidad numerable de elementos, es natural referirse a $(A, \mathcal{P}(A))$ cuando hablemos de A como espacio medible.

Dados S y $\{A_s\}_{s \in S}$, ya se está en disposición de establecer de acuerdo a qué leyes evoluciona el sistema. Al igual que en los procesos de Markov, esto se hace mediante núcleos de Markov, motivo por el cuál estos procesos de decisión reciben este nombre concreto.

[2.1.1] Definición. (Núcleo de Markov).*Dados dos espacios medibles* $(X, \mathcal{A})$ *y* $(Y, \mathcal{B})$, *un núcleo de Markov que parte de* $(X, \mathcal{A})$ *y llega a* $(Y, \mathcal{B})$ *es una función* $\kappa : \mathcal{B} \times X \to [0, 1]$ *tal que*

- *Para todo* $B \in \mathcal{B}$ *fijo,* $\kappa(B|\cdot)$ *es una función* $\mathcal{A}$*-medible;*
- *Para todo* $x \in X$ *fijo,* $\kappa(\cdot|x)$ *es una medida de probabilidad sobre* $(Y,\mathcal{B})$.

En el caso discreto en el que estamos trabajando la primera condición se va a cumplir de manera trivial, siendo relevante solo la segunda. Los núcleos de Markov son los objetos que vamos a dar para especificar las leyes de transición. Así, intuitivamente, utilizaremos núcleos $p(B|(s, a))$ que parten de $(S \times A, \mathcal{P}(S \times A))$ y llegan a $(S, \mathcal{P}(A))$ para encapsular la probabilidad de se que dé el evento B cuando al encontrarnos en el estado s realizamos la acción a (si $B = \{j\}$, abreviaremos $p(\{j\}|(s, a))$ mediante $p(j|(s, a))$ o $p(j|s, a))$. Simbólicamente, en algunos casos escribiremos $p(B|(s, a))$ para pares (s, a) con $a \notin A_s$; en esos casos su valor será irrelevante pues siempre estarán multiplicados por 0 (por comodidad, se puede pensar que se les asigna un valor cualquiera).

Por último, el único elemento intrínseco del modelo que nos falta por describir es la manera en la que se reciben las recompensas. Un método general de establecer esto es utilizando funciones $r : S \times A \times S \to \mathbb{R}$; es decir, la recompensa que se recibe

depende del estado actual, la decisión que se tome y el estado siguiente que se alcance. Una vez más, los valores que tome r en puntos (s,a,j) con (s,a) tal que $a \notin A_s$ serán irrelevantes pues siempre van a ir multiplicados por 0. A partir de todo esto, podemos realizar la siguiente definición.

[2.1.2] Definición. (Proceso de decisión de Markov). *Llamamos proceso de decisión de Markov a la colección de objetos* $\{T, S, \{A_s\}_{s\in S}, p(B|(s,a)), r(s,a,j)\}$

Comentario: Parte de la teoría que se desarrollará en este capítulo se puede aplicar también para el caso en el que tanto las funciones p como r dependen del tiempo. No obstante, se complica la notación y es una extensión que carece de sentido de cara a la segunda parte, donde se trata de estimar el modelo a partir de observaciones en distintos instantes temporales.

2.1.2. Reglas de decisión y políticas

Sobre ese modelo subyacente, el agente habrá de ir tomando decisiones basadas en, a lo sumo, la información que ha acumulado hasta ese instante: el estado actual, los estados en los que se encontró previamente y las acciones que le han llevado hasta la circunstancia actual. Obsérvese que no es necesario incorporar las recompensas que ha recibido hasta esa toma de decisiones, pues éstas son funciones de algo que ya se está considerando. Utilizando el concepto de núcleo de Markov y denotando por H_t al conjunto que encapsula las realizaciones hasta el instante t ($H_0 = S$, $H_1 = H_0 \times A \times S$, ..., $H_t = H_{t-1} \times A \times S$), definimos:

[2.1.3] Definición. *Una regla de decisión en el instante $t \in T$ es un núcleo de Markov q_t que parte de $(H_t, \mathcal{P}(H_t))$ y llega a $(A, \mathcal{P}(A))$ de tal modo que $q_t(\cdot|h_t) = q(\cdot|(s_0, a_0, s_1, a_1, ..., s_t))$ es una probabilidad con soporte en $A_{s_t} \subset A$.*

Es claro que el conjunto de reglas de decisión en el instante t (denotado por D_t^{HR}, donde HR se refiere a *history-dependet randomised*) incluye muchos subconjuntos de interés que a continuación definimos:

[2.1.4] Definición.

- *Las reglas de decisión en el instante t que se pueden escribir como*

$$q_t(\cdot|(s_0, a_0, s_1, a_1, ..., s_t)) = \tilde{q}_t(\cdot|s_t)$$

  (siendo $\tilde{q}$ un núcleo de Markov q que parte de $(S, \mathcal{P}(S))$) se denominan aleatorias de Markov y al conjunto que forman se le denota por D_t^{MR};

- *Las reglas de decisión en el instante t tales que las probabilidades $q_t(\cdot|h_t)$ sobre A están concentradas en un solo punto de A_{s_t} se denominan históricas deterministas y al conjunto que forman se le denota por D_t^{HD};*

- *El subconjunto de las reglas de decisión en el instante t aleatorias de Markov tales que las probabilidades $\tilde{q}_t(\cdot|s_t)$ están concentradas en un punto de A_{s_t} se llaman deterministas de Markov y al conjunto que forman se le denota por D_t^{MD}.*

Está claro que el conjunto D_t^{MD} es el que menos exploración requiere (por ser el más pequeño de los cuatro: D_t^{HR}, D_t^{MR}, D_t^{HD}, D_t^{MD}), y aquel cuyos elementos resultan más sencillos de estimar (pues se pueden determinar completamente mediante una función $d_t : S \to A$ a través de $\tilde{q}_t(a|s_t) = 1 \iff d_t(s_t) = a$). A lo largo de las siguientes secciones, se argumentará que la estrategia óptima - o una arbitrariamente cerca de ésta - podrá encontrarse restringiéndose a buscar en esa clase.

A partir del concepto de regla de decisión se puede definir la idea de política (del inglés *policy*), que no es más que una sucesión que determina en cada instante $t \in T$ la regla de decisión a utilizar.

[2.1.5] Definición. *Una política es un elemento π del conjunto $\Pi = \times_{t=0}^{\infty} D_t^{HR}$. Usando los subconjuntos previamente definidos, se puede hablar de políticas históricas aleatorias, políticas históricas deterministas, políticas aleatorias de Markov y políticas deterministas de Markov como elementos del subconjunto $\Pi^K \subset \Pi$, siendo*

$$\Pi^K = \times_{t=0}^{\infty} D_t^K, \text{ con } K = HD, MR, MD.$$

De este modo, una política es una sucesión del tipo $\pi = (q_0, q_1, q_2, ...)$, siendo q_t núcleos de Markov de H_t a A. En el caso en el que $\pi \in \Pi^{MR}$, a la luz de la definición de D_t^{MR} hay una identificación clara entre D_t^{MR} y $D^{MR} = \{q : \mathcal{P}(A) \times S \to [0,1] |$ q es núcleo de Markov con $q(A_s|s) = 1\}$ para todo $t \in T$; es posible entonces hablar de $\pi \in \Pi^{MR}$ como elemento de $\times_{t=0}^{\infty} D^{MR}$. Entre este tipo de políticas cabe destacar otro subconjunto de interés: el conjunto de políticas que usan la misma regla de decisión en todo instante t o *políticas estacionarias*. Denotaremos a este conjunto - cuyas políticas tienen la forma tienen la forma $\pi = (q, q, q, ...)$ - mediante Π^{SR}. A su vez, al subconjunto de Π^{SR} formado por políticas con reglas de decisión $q_t = q$ deterministas ($q(\cdot|s)$ está concentrada en un punto) lo denominaremos el conjunto de *políticas estacionarias deterministas* o Π^{SD}.

2.1.3. Proceso estocástico inducido

Una vez se nos da un proceso de decisión de Markov, una política y una distribución inicial $\nu : \mathcal{P}(S) \to [0,1]$, es posible ver que toda la dinámica queda completamente determinada. La manera de abordar el estudio de esa evolución es introducir un proceso estocástico canónico que satisface esas condiciones sobre la ley inicial y leyes de transición, y para ello se emplea un modelo probabilístico similar al que se desarrollaba para las cadenas de Markov.

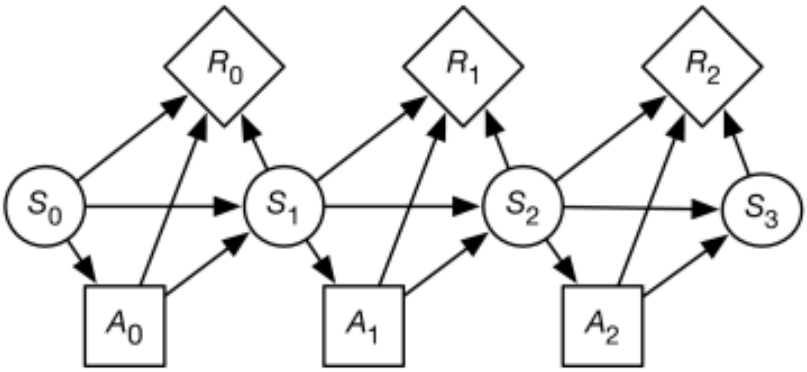


Figura 2.1: Esquema de las interrelaciones presentes en un Proceso de Decisión de Markov. Por simplicidad, en la ilustración solo se considera el caso de las políticas aleatorias de Markov.

La construcción de este proceso comienza considerando el *espacio de realizaciones* Ω. Puesto que no solo estamos interesados en la sucesión de estados que se observa sino también en qué acciones se han tomado en cada instante, tomaremos $\Omega = S \times A \times S \times A \times ... = (S \times A)^{\infty}$. Equipamos Ω con la σ-álgebra producto, que en nuestro caso discreto coincide con $\mathcal{P}(\Omega)$, y definimos las variables aleatorias $\{X_t\}_{t\in T}$, $\{Y_t\}_{t\in T}$ y $\{Z_t\}_{t\in T}$ mediante

$$\begin{aligned} X_t(\omega) :=& X_t((s_0, a_0, s_1, a_1, ...)) = s_t, \\ Y_t(\omega) :=& Y_t((s_0, a_0, s_1, a_1, ...)) = a_t, \\ Z_t(\omega) :=& Z_t((s_0, a_0, s_1, a_1, ...)) = h_t = (s_0, a_0, ..., a_{t-1}, s_t). \end{aligned}$$

Para concluir esta construcción, nos falta asignar una probabilidad al espacio $(\Omega, \mathcal{P}(\Omega))$ que satisfaga las distribuciones condicionales que intuitivamente nos imponen las transiciones $p(\cdot|\cdot)$, la política π y la distribución inicial ν. La manera de llevar a cabo esta última etapa es estándar: asignaremos probabilidades en el álgebra de los conjuntos cilíndricos y aplicaremos el teorema de extensión de Carathéodory. La utilización sucesiva de estos principios básicamente equivale a invocar el teorema de extensión de Daniell-Kolmogorov.

Para cualquier valor de $n \in \mathbb{N}_0$, denotaremos $C_n^s(s_0, a_0, ..., s_n) = \{\omega \in \Omega : X_k(\omega) = s_k,\ k = 0, 1, ..., N$ y $Y_k(\omega) = a_k,\ k = 0, 1, ..., N-1\}$ y $C_n^a(s_0, a_0, ..., s_n, a_n) = \{\omega \in \Omega : X_k(\omega) = s_k,\ k = 0, 1, ..., N$ y $Y_k(\omega) = a_k,\ k = 0, 1, ..., N\}$; es sabido entonces que la familia de conjuntos del tipo $C_n^s(s_0, a_0, ..., s_n)$ y $C_n^a(s_0, a_0, ..., s_n, a_n)$ con $n \in \mathbb{N}_0$ forma un álgebra $\mathcal{C}$. En $\mathcal{C}$, se puede ver - argumentando como en el teorema de Kolmogorov, ver [23] - que la aplicación $\mathbb{P}_\nu^\pi : \mathcal{C} \to [0, 1]$, definida mediante

- $\mathbb{P}_\nu^\pi(C_n^s(s_0, a_0, ..., s_n)) = \nu(s_0)q_1(a_0|h_0)p(s_1|s_0, a_0) \cdots p(s_n|s_{n-1}, a_{n-1})$,

- $\mathbb{P}_\nu^\pi(C_n^s(s_0, a_0, ..., a_n)) = \nu(s_0)q_1(a_0|h_0)p(s_1|s_0, a_0)\cdots q_n(a_n|h_t)$.

es una premedida (de probabilidad) sobre $\mathcal{C}$. Por tanto, por Carathéodory ([21]), concluimos que $\mathbb{P}_\nu^\pi$ se extiende a una medida (de probabilidad) sobre $\sigma(\mathcal{C}) = \mathcal{P}(\Omega)$, a la que denominaremos también $\mathbb{P}_\nu^\pi$. De este modo, nuestro espacio de probabilidad $(\Omega, \mathcal{P}, \mathbb{P}_\nu^\pi)$ dependerá de la política que escojamos a través de la probabilidad que se le asigna, y una política será óptima si la probabilidad que impone en el espacio maximiza un cierto funcional. Respecto al estudio de la influencia de la ley inicial ν en el valor de ese funcional, en el Apéndice A se argumenta que éste es una consecuencia trivial del análisis de los casos degenerados en los que $\nu = \delta_s$.

2.1.4. Ejemplos

Concluimos esta primera sección describiendo dos ejemplos que ilustran el tipo de dinámicas que capturan estos procesos.

[2.1.6] Ejemplo. (Paseo aleatorio en $\mathbb{Z}$ controlado). *Consideremos los instantes temporales $T = \mathbb{N}_0$, el espacio de estados $S = \mathbb{Z}$ y el conjunto de acciones $A_s = A = \{r_i^*, l_e^*, q_u^*\}$. En el estado $k \in \mathbb{Z}$, los núcleos de Markov que encapsulan las distribuciones de transición se definen de la siguiente manera:*

$$\begin{cases} p(j|k, r_i^*) = \frac{1}{2} \text{ si } |j-k-1| = 1 \\ p(j|k, q_u^*) = \frac{1}{2} \text{ si } |j-k| = 1 \\ p(j|k, l_e^*) = \frac{1}{2} \text{ si } |j-k+1| = 1 \end{cases}$$

Aceptamos que la ditribución inicial es una ν cualquiera, y suponemos que las recompensas son del tipo $r(k, a, j) = e^{-k^2}$. Es intuitivo que una política interesante en estas condiciones puede ser la que viene dada por siguiente regla de decisión de Markov en todo instante

$$\begin{cases} q(r_i^*|k) = 1 \text{ y } q(q_u^*|k) = q(l_e^*|k) = 0 \text{ si } k < 0 \\ q(q_u^*|k) = 1 \text{ y } q(r_i^*|k) = q(l_e^*|k) = 0 \text{ si } k = 0 \\ q(l_e^*|k) = 1 \text{ y } q(r_i^*|k) = q(q_u^*|k) = 0 \text{ si } k > 0 \end{cases}$$

En esta situación, es sencillo notar que las v.a. $\{X_t\}_{t\in T}$ satisfacen una dinámica de cadena de Markov que, eso sí, no coincide con el paseo aleatorio en $\mathbb{Z}$.

[2.1.7] Ejemplo. *Dado un espacio de estados S y un conjunto de acciones posibles para cada estado $\{A_s\}_{s\in S}$, supongamos que queremos establecer un modelo como se suele hacer en sistemas dinámicos discretos. Para ello, consideraríamos una fuente de ruido (un conjunto de v.a. independientes $\{w_t\}_{t\in T}$ con misma distribución que toman valores en W) y damos una $f : S \times A \times W$ de tal modo que*

$$s_{t+1} = f(s_t, a_t, w_t).$$

Este tipo de modelos se pueden capturar con los procesos de decisión de Markov, donde $p(\cdot|s,a)$ vendría dado mediante

$$p(j|s,a) = \mathbb{P}(\{w \in W : f(s,a,w) = j\})$$

De manera similar se traduciría al lenguaje de los procesos de decisión de Markov la política escogida.

El tipo de modelos presentados en el ejemplo anterior son, de hecho, igual de generales que los procesos de decisión de Markov: en efecto, dado el modelo expresado de una forma (procesos de decisión de Markov o sistema dinámico), siempre es posible encontrar un modelo equivalente en la otra formulación (ver [22]).

2.2. El valor de una política, la función valor y las políticas de Markov

Una vez se ha establecido el modelo subyacente, hay que proceder a definir un criterio de acuerdo al cuál se pueda determinar la bondad de una política o, al menos, su calidad en comparación con otras. En esta tarea, es claro que las recompensas $r(s,a,j)$ van a desempeñar un rol crucial, aunque no es tan evidente encontrar - dado un problema - la manera de ponderar adecuadamente las recompensas recibidas en los distintos instantes temporales.

2.2.1. El criterio de la recompensa total descontada esperada y la función valor asociada

Como ya se comentó, existen muchas formas diferentes de realizar esta asignación (centrarse en la recompensa media, en la recompensa total acumulada, en el ritmo de crecimiento de esa acumulación,...). Cada uno de estos criterios da lugar a problemas de control que, si bien se abordan con enfoques similares, requieren razonamientos particulares. Con el objeto de no extender en exceso la exposición, aquí nos centraremos en una de las maneras de valorar una política más habituales: la recompensa total descontada esperada. Ésta es una ponderación que se suele asociar a problemas en los que los instantes representan épocas temporales y el agente prioriza ganancias actuales frente a futuras (como ocurre de forma ubicua en problemas económicos); no obstante, también surge en ocasiones de otro tipo de naturaleza (por ejemplo, procesos sometidos a una parada aleatoria que sigue una distribución geométrica, ver [31])

Recordando que $\mathbb{P}_\nu^\pi$ denotaba la probabilidad que induce en $(\Omega, \mathcal{P}(\Omega))$ la política π junto con la ley inicial ν (cuando ν esté concentrada en un estado s, utilizaremos $\mathbb{P}_s^\pi$), usaremos $\mathbb{E}_\nu^\pi$ ($\mathbb{E}_s^\pi$ si $\nu = \delta_s$) para denotar el operador esperanza asociado a esa probabilidad. Introducimos también el espacio vectorial $\mathcal{V} = \{V \in \mathbb{R}^S : \|V\|_\infty < \infty\}$,

que es claramente un espacio de Banach por ser isometricamente isomorfo a un subespacio cerrado de l^∞. Con esta notación, definimos:

[2.2.8] Definición. (Valor de una política). *La recompensa total descontada esperada de una política π para un factor de descuento $\lambda \in [0,1)$ es una función de $\mathcal{V}$ tal que*

$$V_\lambda^\pi(s) = \lim_{N\to\infty} \mathbb{E}_s^\pi \left[\sum_{t=0}^{N} \lambda^t r(X_t, Y_t, X_{t+1})\right]$$

De cara a que la definición sea buena para cualquier $\pi \in \Pi^{HR}$, es necesario que exista una constante C_π tal que $|V_\lambda^\pi(s)| \leq C_\pi$ para todo $s \in S$. Una manera habitual de garantizar que se satisface esta condición - y que no es restrictiva con vistas a los modelos computacionales - es realizar la siguiente asunción.

[2.2.9] Asunción. *Las recompensas $r : S \times A \times S \to \mathbb{R}$ satisfacen la siguiente condición de acotación para algún $M \in \mathbb{R}_+$.*

$$\sup_{(s,a,j)\in S\times A\times S} |r(s,a,j)| \leq M$$

En ese caso, se tiene que para todo $N \in \mathbb{N}$, $|\sum_{t=0}^{N} \lambda^t r(X_t, Y_t, X_{t+1})| \leq \sum_{t=0}^{N} \lambda^t M \leq \frac{M}{1-\lambda}$, por lo que por el Teorema de la Convergencia Dominada nos permite confirmar que efectivamente $V_\lambda^\pi \in \mathcal{V}$ para toda π pues

$$|V_\lambda^\pi(s)| = \left|\mathbb{E}_s^\pi \left[\sum_{t=0}^{\infty} \lambda^t r(X_t, Y_t, X_{t+1})\right]\right| \leq \frac{M}{1-\lambda}.$$

A partir de estas funciones V_λ^π ya es posible determinar la calidad de las políticas para cada condición inicial, y se puede establecer un orden parcial según el cuál $V_\lambda^{\pi_1} \leq V_\lambda^{\pi_2} \iff V_\lambda^{\pi_1}(s) \leq V_\lambda^{\pi_2}(s)$ para todo $s \in S$. La presencia de este orden sugiere que la definición que se presenta a continuación puede ser útil al menos como referencia de la bondad de una política. Sin embargo, como veremos en la Sección 2.3, el rol que juega es mucho más profundo que eso.

[2.2.10] Definición. (Función valor). *La función valor del proceso de decisión de Markov (o, brevemente, la función valor) se define del siguiente modo*

$$V_\lambda^*(s) = \sup_{\pi\in\Pi^{HR}} V_\lambda^\pi(s).$$

Comentario: En la definición anterior, el supremo se toma puntualmente, de tal modo que es trivial que $V_\lambda^\pi(s) \leq V_\lambda^*(s)$ para todo $s \in S$ y cualquier $\pi \in \Pi^{HR}$. Dado que $|V^\pi(s)| \leq \frac{M}{1-\lambda}$, es claro que $V_\lambda^* \in \mathcal{V}$.

Comentario: La relevancia de la función valor no es una particularidad de la teoría de procesos de decisión de Markov. Constituye un pilar básico de la teoría de control óptimo moderna en su aplicación a todo tipo de modelos (desde discretos deterministas a continuos estocásticos). Cuando sea pertinente, se enfatizará en la universalidad de algunas técnicas que vamos a exponer.

La función valor representa una cota para la calidad de las políticas y nuestro objetivo es, en primer lugar, conocerla y, en segundo lugar, encontrar un algoritmo para o alcanzarla o acercarse arbitrariamente a ella. La siguiente subsección nos detalla la relación de V_λ^* con el subconjunto formado por las políticas de Markov.

2.2.2. Las políticas de Markov

Es evidente que el subconjunto Π^{MR} es mucho más pequeño y manejable a la hora de tratarlo que Π^{HR}. Afortunadamente, es suficiente para poder aproximar la cota superior V_λ^* con precisión arbitraria.

[2.2.11] Teorema. *Dada una política $\pi \in \Pi^{HR}$, existe para cada $s \in S$ una política $\pi' \in \Pi^{MR}$ tal que $V_\lambda^\pi(s) = V_\lambda^{\pi'}(s)$. De este modo se tiene que*

$$V_\lambda^*(s) = \sup_{\pi \in \Pi^{HR}} V_\lambda^\pi(s) = \sup_{\pi \in \Pi^{MR}} V_\lambda^\pi(s).$$

[2.2.11] *Demostración.* La última afirmación es una consecuencia trivial de la primera, por lo que solo mostraremos ésta. Procedemos en dos pasos: primero vemos que para $\pi \in \Pi^{HR}$ existe un $\pi' \in \Pi^{MR}$ tal que en todo $t \in T$ y con s fijo se cumple que $\mathbb{P}_s^\pi(\{X_t = j, Y_t = a, X_{t+1} = k\}) = \mathbb{P}_s^{\pi'}(\{X_t = j, Y_t = a, X_{t+1} = k\})$, y a continuación usamos esto para llegar a $V_\lambda^\pi(s) = V_\lambda^{\pi'}(s)$. La idea original se le atribuye a Derman y Strauch ([12]) y se puede extender a contextos más generales.

Paso 1. Definimos $\pi' = (q_0', q_1', ...)$ de tal modo que en los valores de j tales que $\mathbb{P}_s^\pi(\{X_t = j\}) \neq 0$ se cumpla

$$q_t'(a|j) = \mathbb{P}_s^\pi(\{Y_t = a\}|\{X_t = j\}).$$

Con esta definición, q_t' es núcleo de Markov de S a A con $q_t'(\cdot|j)$ soportada en A_j. Procedemos ahora por inducción. $\mathbb{P}_s^\pi(\{X_t = j, Y_t = a, X_{t+1} = k\}) = \mathbb{P}_s^{\pi'}(\{X_t = j, Y_t = a, X_{t+1} = k\})$ es cierto si $t = 0$; aceptando que se cumple para $t = n$, vemos

que

$$\begin{aligned}\mathbb{P}_s^{\pi}(\{X_n = j\}) &= \sum_{k\in S}\sum_{a\in A_k} \mathbb{P}_s^{\pi}(\{X_n = j, Y_{n-1} = a, X_{n-1} = k\}) = \\ &= \sum_{k\in S}\sum_{a\in A_k} \mathbb{P}_s^{\pi'}(\{X_n = j, Y_{n-1} = a, X_{n-1} = k\}) = \\ &= \mathbb{P}_s^{\pi'}(\{X_n = j\}).\end{aligned}$$

Por tanto,

$$\begin{aligned}\mathbb{P}_s^{\pi}(\{X_n = j, Y_n = a, X_{n+1} = k\}) &= \mathbb{P}_s^{\pi}(\{X_{n+1} = k\}|\{Y_n = a, X_n = j\})\mathbb{P}_s^{\pi}(\{X_n = j, Y_n = a\}) = \\ &= p(k|j,a)\mathbb{P}_s^{\pi}(\{Y_n = a\}|\{X_n = j\})\mathbb{P}_s^{\pi}(\{X_n = j\}) = \\ &= p(k|j,a)q'_n(a|j)\mathbb{P}_s^{\pi'}(\{X_n = j\}) = \\ &= \mathbb{P}_s^{\pi'}(\{X_n = j, Y_n = a, X_{n+1} = k\}).\end{aligned}$$

Paso 2. Se tiene, por el Teorema de la Convergencia Dominada, que

$$\begin{aligned}V_\lambda^{\pi}(s) &= \sum_{t=0}^{\infty}\sum_{k\in S}\sum_{a\in A}\sum_{s\in S} \lambda^t r(s,a,k)\mathbb{P}_s^{\pi}(\{X_{t+1} = k, Y_t = a, X_t = s\}) = \\ &= \sum_{t=0}^{\infty}\sum_{k\in S}\sum_{a\in A}\sum_{s\in S} \lambda^t r(s,a,k)\mathbb{P}_s^{\pi'}(\{X_{t+1} = k, Y_t = a, X_t = s\}) = V_\lambda^{\pi'}(s).\end{aligned}$$

□

Comentario: Si tomamos un $\pi \in \Pi^{HR}$ y dos estados $s \neq j$, la política $\pi'_s \in \Pi^{MR}$ tal que para $s \in S$ cumple que $V_\lambda^{\pi}(s) = V_\lambda^{\pi'_s}(s)$ puede no coincidir con la política $\pi'_j \in \Pi^{MR}$ tal que $V_\lambda^{\pi}(j) = V_\lambda^{\pi'_j}(j)$. De este modo, el teorema anterior no descarta la posibilidad de que exista una única π tal que $V_\lambda^* = V_\lambda^{\pi}$ y que ésta cumpla $\pi \in \Pi^{HR} \setminus \Pi^{MR}$.

Sin embargo, aunque este teorema no nos permita concluir categóricamente que basta centrar nuestra atención en los elementos de Π^{MR}, si que nos indica que son suficientes al menos de cara al cálculo de V_λ^*. Dado que esta tarea copará el análisis de la próxima sección, merece la pena realizar una pequeña discusión previa que nos permita manipular las políticas de Markov con mayor comodidad y entendimiento; con este fin, comenzamos notando un hecho *a priori* intuitivo.

[2.2.12] Teorema. *El proceso estocástico inducido al emplear una $\pi \in \Pi^{MR}$ es una cadena de Markov para las v.a. $\{X_t\}_{t\in T}$.*

[2.2.12] *Demostración.* Utilizando la identificación anteriormente mencionada, un elemento $\pi \in \Pi^{MR}$ se puede escribir como $\pi = (q_0, q_1, ...)$, siendo los q_i núcleos de Markov de S a A. Para mostrar que las v.a. $\{X_t\}_{t\in T}$ forman una cadena de Markov, basta ver ([8]) que para todo $t \in T$

$$\mathbb{P}^\pi_\nu(\{X_{t+1} = s_{t+1}\}|\{X_k = s_k,\, k = 0, 1, ..., t\}) = \mathbb{P}^\pi_\nu(\{X_{t+1} = s_{t+1}\}|\{X_t = s_t\}).$$

Esto es trivial, pues ambos términos son iguales a $\sum_{a\in A} q_t(a|s_t)p(s_{t+1}|s_t, a)$.

□

Del teorema anterior, se deduce también que el núcleo de Markov de dicha cadena en el instante t vendrá dado por $P_{q_t}(s_{t+1}|s_t) = \sum_{a\in A} q_t(a|s_t)p(s_{t+1}|s_t, a)$. Éstos, como es habitual en procesos de Markov, serán utilizados en ocasiones como operadores de $\mathbb{R}^S$ en $\mathbb{R}^S$ definidos mediante

$$\begin{aligned} P_{q_t} &: \mathbb{R}^S \to \mathbb{R}^S \\ & V(s) \mapsto \sum_{j\in S} P_{q_t}(j|s)V(j) \end{aligned}$$

Usando las propiedades que satisface $P_{q_t}(j|s)$, es fácil ver que en concreto esa definición es buena considerándola como operador de $\mathcal{V}$ en $\mathcal{V}$, y que en ese caso $\|P_{q_t}\|_{\mathcal{V}\to\mathcal{V}} = 1$ ($\|\cdot\|_{\mathcal{V}\to\mathcal{V}}$ es la norma de operadores). Análogamente, en lo referido a notación, hablaremos también de $P^\pi_t(j|s) = \mathbb{P}^\pi_s(\{X_t = j\})$, que cuando se interprete como operador de $\mathcal{V}$ en $\mathcal{V}$ satisface,

$$P^\pi_t = P_{q_0} \circ P_{q_1} \circ \cdots \circ P_{q_{t-1}}. \tag{2.1}$$

Una vez más, es sencillo ver que $\|P^\pi_t\|_{\mathcal{V}\to\mathcal{V}} = 1$. Por último, para terminar esta aclaración de nomenclatura, introducimos

$$r_{q_t}(X_t) = \mathbb{E}^\pi_\nu \left[r(X_t, A_t, X_{t+1})|\sigma(X_t)\right],$$

que en este contexto es fácil ver que se reduce a la expresión

$$r_{q_t}(X_t) = \sum_{j\in S}\sum_{a\in A} r(X_t, a, j)q_t(a|X_t)p(j|X_t, a).$$

No es difícil notar ahora que una manera de escribir $\mathbb{E}^\pi_s [r(X_t, A_t, X_{t+1})]$ es

$$\begin{aligned} \mathbb{E}^\pi_s [r(X_t, A_t, X_{t+1})] &= \mathbb{E}^\pi_s [\mathbb{E}^\pi_s [r(X_t, A_t, X_{t+1})|\sigma(X_t)]] = \\ &= \mathbb{E}^\pi_s [r_{q_t}(X_t)] = \sum_{j\in S} \mathbb{P}^\pi_s(\{X_t = j\})r_{q_t}(j) = \sum_{j\in S} P^\pi_t(j|s)r_{q_t}(j). \end{aligned}$$

De este modo, interpretando P_t^π como operador, se pueden obtener expresiones para V_λ^π como la siguiente:

$$V_\lambda^\pi(s) = \sum_{t=0}^{\infty} \lambda^t \mathbb{E}_s^\pi \left[r(X_t, A_t, X_{t+1}) \right] = \sum_{t=0}^{\infty} \lambda^t (P_t^\pi r_{q_t})(s). \tag{2.2}$$

2.3. El principio de optimalidad y su aplicación en el diseño de políticas

Nos encontramos ya en disposición de exponer el procedimiento para encontrar políticas óptimas (o arbitrariamente cercanas a la cota superior) para procesos de decisión de Markov. Es en este punto dónde el principio de optimalidad y la función V_λ^* van a desempeñar un papel esencial. El lector no debe confundir la particularidad de los resultados que se van a presentar aquí con una posible falta de generalidad del enfoque a seguir; como ya se comentó anteriormente, la esencia de la teoría que se va a desarrollar permite abordar con éxito una gran variedad de problemas de teoría de control óptimo con diversos tipos de modelos subyacentes.

Esta manera paradigmática de proceder se basa en la siguiente secuencia de razonamientos. En primer lugar, nos fijamos en las relaciones temporales que presenta el proceso y el criterio de optimización asociado; en nuestro caso, esto se traduce en lo siguiente

- Para un proceso con una política de Markov $\pi = (q_0, q_1, ...)$ y ley inicial $\nu = \delta_s$, parece realista conjeturar que $\{\tilde{X}_t\}_{t \in T} = \{X_{t+1}\}_{t \in T}$ es un proceso con política $\tilde{\pi} = (\tilde{q}_0, \tilde{q}_1, ...) = (q_1, q_2, ...)$ y ley inicial $\nu(j) = \mathbb{P}_s^\pi(\{X_1 = j\})$.

- $V_\lambda^\pi(s) = \sum_{t=0}^{\infty} \lambda^t \mathbb{E}_s^\pi \left[r_{q_t}(X_t) \right] = r_{q_0}(s) + \lambda \sum_{t=1}^{\infty} \lambda^{t-1} \mathbb{E}_s^\pi \left[\mathbb{E}_s^\pi \left[r_{q_t}(X_t) | \sigma(X_1) \right] \right]$.

Por tanto, observamos que una manera de descomponer la recompensa total descontada esperada es básicamente considerar la suma de la recompensa que se va obtener en ese instante más la recompensa total descontada esperada del resto del proceso, que a su vez parece seguir una dinámica relacionada con la política original. Si escogemos $\{X_{t+1}\}_{t \in T}$ para que se comporten de manera óptima (o estén al menos *muy cerca*), veríamos intuitivamente que

$$V_\lambda^\pi(s) = \sum_{t=0}^{\infty} \lambda^t \mathbb{E}_s^\pi \left[r_{q_t}(X_t) \right] \approx r_{q_0}(s) + \lambda \sum_{j \in S} P_{q_0}(j|s) V_\lambda^*(j).$$

De este modo, la ley de decisión q_0 que más capacidad tiene de conseguir recompensa total descontada es aquella que cumple que para todo $s \in S$

$$r_{q_0}(s) + P_{q_0}(j|s)V_\lambda^*(s) = \max_q\{r_q(s) + P_q(j|s)V_\lambda^*(s)\}.$$

Vamos a ver que, efectivamente, esta línea argumental nos lleva a buen puerto. No obstante, como observamos en la igualdad anterior, para utilizar este método de cara a la síntesis de políticas es necesario encontrar previamente un método de cálculo para la función V_λ^*. Estas tareas son las que llevaremos a cabo con rigor en las próximas subsecciones.

2.3.1. El principio de optimalidad y la ecuación de Bellman

Comenzamos abordando la labor de encontrar un algoritmo para conocer la los valores que toma la función V_λ^*. Para ello, nos centraremos en la observación recién realizada sobre la descomposición de la función valor en dos sumandos y utilizaremos el principio de optimalidad presentado en la introducción.

Principio de optimalidad. *Una sucesión de decisiones óptima satisface que, dados un estado y un instante inicial cualquiera, las acciones de los instantes posteriores constituyen una manera de actuar óptima para el estado resultante tras la primera decisión.*

La transcripción matemática de este enunciado requiere la observación previa de que las políticas de Markov (usando notación introducida previamente, $\pi \in \Pi^{MR} = (D^{MR})^\infty$) se pueden expresar como un elemento $\pi = (q, \pi')$ de $D^{MR} \times (D^{MR})^\infty$. Bajo esta notación, la ecuación (2.2) se escribe del siguiente modo para una política cualquiera

$$\begin{aligned} V_\lambda^\pi(s) &= \sum_{t=0}^{\infty} \lambda^t (P_t^\pi r_{q_t})(s) = r_{q_0}(s) + \lambda \sum_{t=0}^{\infty} \lambda^t (P_{t+1}^\pi r_{q_{t+1}})(s) = \\ &= r_{q_0}(s) + \lambda P_{q_0} \left(\sum_{t=0}^{\infty} \lambda^t (P_t^{\pi'} r_{q_t'}) \right)(s) = r_{q_0}(s) + \lambda \left(P_{q_0} V_\lambda^{\pi'} \right)(s), \end{aligned}$$

donde en la tercera igualdad se ha empleado el hecho expuesto en (2.1).

Habiendo observado esto, suponemos ahora que para todo estado $s \in S$ es posible alcanzar una política óptima π_s^* de tal modo que $V_\lambda^*(s) = V_\lambda^{\pi_s^*}(s)$. Entonces, para que la política anterior sea óptima para $s \in S$ ha de satisfacer dos condiciones:

1. Por el principio de optimalidad, se ha de cumplir que $V_\lambda^{\pi'}(s) = V_\lambda^*(s)$, por lo que $\left(P_{q_0} V_\lambda^{\pi'} \right)(s) = \sum_{j \in S} P_{q_0}(j|s) V_\lambda^\pi(j) = \sum_{j \in S} P_{q_0}(j|s) V_\lambda^*(j)$;

2. La primera ley de decisión ha de ser también óptima, por lo que $r_{q_0}(s) + \lambda \left(P_{q_0} V_\lambda^{\pi'} \right)(s) = \text{máx}_q \{ r_q(s) + \lambda \left(P_q V_\lambda^{\pi'} \right)(s) \}$.

Juntando ambos razonamientos, llegamos a lo que se conoce como ecuación de Bellman. Lo argumentado hasta ahora nos sugiere que ésta es la relación que ha de satisfacer V_λ^*.

[2.3.13] Definición. *Denominamos ecuación de Bellman al siguiente problema planteado en el espacio de funciones $\mathcal{V}$.*

$$V(s) = \sup_{q \in D^{MR}} \{ r_q(s) + \lambda \sum_{j \in S} P_q(j|s) V(j) \}.$$

Comentario: En la ecuación de Bellman se ha de tomar supremo pues en principio no sabemos si hablar de máximo tiene sentido. En efecto, no es difícil encontrar ejemplos prácticos en los que ese supremo no se alcanza; no obstante, como veremos en los próximos párrafos, se puede mostrar con rigor que la ecuación de Bellman que acabamos de derivar heurísticamente aplica también en estas circunstancias.

Comentario: Debido al horizonte infinito en el que se plantea el problema que estamos trabajando aquí, no aparece ni en la función valor ni - por tanto - en la ecuación de Bellman ninguna dependencia temporal. Los problemas en los que nuestro proceso se termina en un instante T sí que dan lugar a dependencias temporales, definiéndose ahora la función valor de tal modo que $V^* = V^*(s,t)$. En estas situaciones se obtienen ecuaciones de Bellman similares pero con una relación que permite obtener los valores de $V^*(s,t)$ a partir de los de $V^*(s,t+1)$. La manera de abordar estas circunstancias es mediante un método de resolución recursiva hacia atrás: se parte de los valores $V^*(s,T)$ y a partir de ahí se calculan iterativamente los demás. Este tipo de técnicas - fuertemente asociadas en la literatura al principio de optimalidad y al propio Bellman, y con aplicaciones en ciencias de la computación ([9]) - están relacionadas pero no coinciden con lo que tratamos aquí. El lector interesado en estos fenómenos motivados por la finitud del número de instantes temporales es remitido a [3] y [31].

Comenzamos el análisis de la ecuación de Bellman notando que, a efectos de ésta, es suficiente trabajar con las leyes de decisión de Markov deterministas.

[2.3.14] Proposición. *Para cualquier $V \in \mathcal{V}$ y $\lambda \in [0,1)$, se tiene que*

$$\sup_{q \in D^{MR}} \{ r_q(s) + \lambda \sum_{j \in S} P_q(j|s) V(j) \} = \sup_{q \in D^{MD}} \{ r_q(s) + \lambda \sum_{j \in S} P_q(j|s) V(j) \}.$$

[2.3.14] *Demostración.* Dado que $D^{MD} \subset D^{MR}$, se tiene trivialmente que $\sup_{q\in D^{MR}}\{r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j)\} \geq \sup_{q\in D^{MD}}\{r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j)\}$. Para ver la desigualdad contraria, notamos que para todo $q \in D^{MR}$

$$r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j) = \sum_{j\in S}\sum_{a\in A} r(s,a,j)q(a|s)p(j|s,a) + \lambda\sum_{j\in S}\sum_{a\in A} q(a|s)p(j|s,a)V(j) =$$

(2.3)

$$= \sum_{a\in A} q(a|s)\left[\sum_{j\in S} r(s,a,j)p(j|s,a) + \lambda\sum_{j\in S} p(j|s,a)V(j)\right].$$

Las leyes de decisión $q \in D^{MD}$ son tales que para cada s están concentradas en un elemento $a_q(s)$ de A_s, por lo que

$$q \in D^{MD} \implies r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j) = \sum_{j\in S} r(s,a_q(s),j)p(j|s,a_q(s))+\lambda\sum_{j\in S} p(j|s,a_q(s))V(j).$$

De este modo,

$$\sup_{q\in D^{MD}}\{r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j)\} = \sup_{a\in A_s}\left\{\sum_{j\in S} r(s,a,j)p(j|s,a) + \lambda\sum_{j\in S} p(j|s,a)V(j)\right\}.$$

Esta igualdad, junto con (2.3), nos permite concluir que

$$\sup_{q\in D^{MR}}\{r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j)\} \leq \sup_{q\in D^{MD}}\{r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j)\}.$$

□

Comentario: De ahora en adelante usaremos la notación

$$\sup_{q\in D^{MR}}\{r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j)\} = \sup_{q\in D^{MD}}\{r_q(s)+\lambda\sum_{j\in S}P_q(j|s)V(j)\} = \mathcal{L}V,$$

donde $\mathcal{L} : \mathcal{V} \to \mathcal{V}$ es un operador continuo no lineal al que llamaremos *operador de Bellman*.

Utilizando la proposición anterior, es posible probar que, entre otras cosas, si la ecuación de Bellman tiene solución ésta necesariamente ha de coincidir con V^*_λ.

[2.3.15] Teorema. *Supongamos que existe una $V \in \mathcal{V}$ tal que*

1. $V \geq \mathcal{L}V$; *entonces,* $V \geq V_\lambda^*$;
2. $V \leq \mathcal{L}V$; *entonces,* $V \leq V_\lambda^*$;
3. $V = \mathcal{L}V$; *entonces,* $V = V_\lambda^*$.

[2.3.15] *Demostración.*

Punto 1. Comenzamos tomando una política aleatoria de Markov cualquiera $\pi = (q_0, q_1, ...)$. Dado que $V(s) \geq \sup_{q \in D^{MR}}\{r_q(s) + \lambda(P_{q_t}V)(s)\}$, es evidente que para cualquier q_t se tiene que

$$V(s) \geq r_{q_t}(s) + \lambda(P_{q_t}V)(s) \tag{2.4}$$

Es fácil ver que, debido a la positividad de $P_{q_t}(j|s)$, si $V_1 \geq V_2$ entonces se tiene que

$$(P_{q_t}V_1)(s) = \sum_{j \in S} P_{q_t}(j|s)V_1(j) \geq \sum_{j \in S} P_{q_t}(j|s)V_2(j) = (P_{q_t}V_2)(s).$$

Por tanto, aplicando (2.4) iterativamente, vemos que

$$\begin{aligned} V(s) \geq & r_{q_0}(s) + \lambda(P_{q_0}V)(s) \geq r_{q_0}(s) + \lambda(P_{q_0}r_{q_1})(s) + \lambda^2(P_{q_0}P_{q_1}V)(s) \geq ... \geq \\ \geq & r_{q_0}(s) + \lambda(P_{q_0}r_{q_1})(s) + ... + \lambda^{n-1}(P_{q_0}P_{q_1} \cdots P_{q_{n-2}}r_{q_{n-1}})(s) + \lambda^n(P_{q_0}P_{q_1} \cdots P_{q_{n-1}}V)(s) = \\ = & r_{q_0}(s) + \lambda(P_1^\pi r_{q_1})(s) + ... + \lambda^{n-1}(P_{n-1}^\pi r_{q_{n-1}})(s) + \lambda^n(P_n^\pi V)(s). \end{aligned}$$

Al final de la Subsección 2.2.2 argumentamos que $V_\lambda^\pi(s) = \sum_{t=0}^\infty \lambda^t(P_t^\pi r_{q_t})(s)$, por lo que ahora observamos, restando esa igualdad en la desigualdad anterior, que

$$V(s) - V_\lambda^\pi(s) \geq \lambda^n(P_n^\pi V)(s) - \sum_{t=n}^\infty \lambda^t(P_t^\pi r_{q_t})(s).$$

Utilizando el hecho de que $|r_{q_k}(s)| \leq M$ para cualquier $k \in \mathbb{Z}$ y recordando que $\|P_k^\pi\|_{\mathcal{V} \to \mathcal{V}} = 1$ para $k \in \mathbb{N}_0$ y $\pi \in \Pi^{MR}$ arbitraria, notamos que

- $|\lambda^n(P_n^\pi V)(s)| \leq \lambda^n \|V\|_\infty$,
- $\left|\sum_{t=n}^\infty \lambda^t(P_t^\pi r_{q_t})(s)\right| \leq \sum_{t=n}^\infty \lambda^t M = \frac{M\lambda^n}{1-\lambda}$.

De este modo,

$$V(s) - V_\lambda^\pi(s) \geq -\lambda^n\|V\|_\infty - \frac{M\lambda^n}{1-\lambda}.$$

Tomando límite cuando $n \to \infty$, concluimos que $V(s) - V_\lambda^\pi(s) \geq 0$; como hemos razonado para cualquier $s \in S$, $V - V_\lambda^\pi \geq 0$, y dado que $\pi \in \Pi^{MR}$ era arbitraria se

tiene que $V \geq V_\lambda^*$.

Punto 2. Sea ahora $V \leq \mathcal{L}V$. Denotaremos por e la función $\mathbb{R}^S$ tal que $e(s) = 1$ en todo $s \in S$. Entonces, si fijamos un $\epsilon > 0$, en base a lo establecido en la Proposición 2.3.14 se puede encontrar un $\tilde{q} \in D^{MD}$ (siendo $a_{\tilde{q}}(s)$ el elemento de A_s tal que $\tilde{q}(a|s) = 1$) que satisface que

$$V \leq \mathcal{L}V \leq r_{\tilde{q}} + \lambda(P_{\tilde{q}}V) + \epsilon e. \tag{2.5}$$

Considerando ahora la política dada por $\tilde{\pi} = (\tilde{q}, \tilde{q}, ...)$, vemos que

$$\begin{aligned} V_\lambda^{\tilde{\pi}} &= \sum_{t=0}^{\infty} \lambda^t (P_t^{\tilde{\pi}} r_{\tilde{q}}) = r_{\tilde{q}} + \lambda \sum_{t=0}^{\infty} \lambda^t (P_{t+1}^{\tilde{\pi}} r_{\tilde{q}}) = \\ &= r_{\tilde{q}} + \lambda \left(P_{\tilde{q}} \left[\sum_{t=0}^{\infty} \lambda^t (P_t^{\tilde{\pi}} r_{\tilde{q}}) \right] \right) = r_{\tilde{q}} + \lambda (P_{\tilde{q}} V_\lambda^{\tilde{\pi}}). \end{aligned} \tag{2.6}$$

Restando esta igualdad de (2.5), llegamos a

$$(I - \lambda P_{\tilde{q}})(V - V_\lambda^{\tilde{\pi}}) \leq \epsilon e.$$

Observamos ahora que el operador $(I - \lambda P_{\tilde{q}}) = -\lambda(P_{\tilde{q}} - \frac{1}{\lambda} I)$ es invertible, puesto que el espectro de $P_{\tilde{q}}$ está contenido en $[-\|P_{\tilde{q}}\|_{\mathcal{V}\to\mathcal{V}}, \|P_{\tilde{q}}\|_{\mathcal{V}\to\mathcal{V}}] = [-1, 1]$ (ver [5]). Por otro lado, razonando como en la ecuación (2.6), vemos que fijando un vector $Y \in \mathcal{V}$ (siendo $\tilde{\pi} \in \Pi^{SR}$), la única solución de la ecuación

$$(I - \lambda P_{\tilde{q}})X = Y$$

coincide con la función $V_\lambda^{\tilde{\pi}}$ para un proceso de decisión de Markov tal que $r_{\tilde{q}}(s) = Y(s)$; bajo esta interpretación, es obvio que si $Y \geq 0$ entonces se tiene que $X \geq 0$. Por tanto, observando que

$$(I - \lambda P_{\tilde{q}})(V - V_\lambda^{\tilde{\pi}}) \leq \epsilon e = (I - \lambda P_{\tilde{q}}) \left(\frac{\epsilon}{1-\lambda} e \right)$$

vemos que

$$(I - \lambda P_{\tilde{q}})(V - V_\lambda^{\tilde{\pi}} - \frac{\epsilon}{1-\lambda} e) \leq 0 \implies V - V_\lambda^{\tilde{\pi}} - \frac{\epsilon}{1-\lambda} e \leq 0.$$

Para terminar, solo falta ver que

$$V \leq V_\lambda^{\tilde{\pi}} + \frac{\epsilon}{1-\lambda} e \leq V_\lambda^* + \frac{\epsilon}{1-\lambda} e.$$

Como ϵ es arbitrario, hemos concluido.

Punto 3. Se deduce de manera trivial de los Puntos 1 y 2.

□

Comentario: El teorema anterior no me dice que haya solución a la ecuación de Bellman; solamente me afirma que, de existir, ésta es única pues necesariamente ha de coincidir V_λ^* (la función V_λ^* sí que es claramente única, pero todavía no sabemos si resuelve la ecuación de Bellman).

Comentario: Al aplicar este argumento en el caso continuo determinista se obtienen, para la versión correspondiente de la ecuación de Bellman (que en esa situación recibe el nombre de Hamilton-Jacobi-Bellman), unas relaciones muy similares a las que establece el Teorema 2.3.15. Sobre ese tipo de condiciones se fundamenta una cantidad significativa de matemática: dan lugar a lo que se conoce como soluciones de viscosidad, concepto introducido en los años 80 por M.G. Crandall y P.-L. Lions ([10]) y que permitió abordar con éxito ecuaciones del tipo de Hamilton-Jacobi. Aunque en su origen se idearon independientemente, pronto se identificó la relación entre ambas teorías, dando lugar a un tratamiento unificado y muy elegante. El lector interesado es remitido a [17] para un tratamiento compacto y breve, y a [7] y [2] para textos más detallados. Además, condiciones de este tipo también juegan un papel fundamental en la teoría de control óptimo de sistemas estocásticos continuos (ver [20]).

De este modo, el teorema anterior nos confirma que efectivamente la ecuación de Bellman es la relación matemática adecuada para intentar calcular la función V_λ^*. Con todo lo discutido hasta ahora, solo nos falta mostrar que dicha ecuación tiene al menos una solución, lo cuál se deduce fácilmente observando que el operador $\mathcal{L}$ es contractivo.

[2.3.16] Teorema. *El operador $\mathcal{L}$ es contractivo, y la ecuación*

$$V = \mathcal{L}V$$

tiene solución única en $\mathcal{V}$.

[2.3.16] *Demostración.* El espacio $\mathcal{V}$ es de Banach (puesto que, como ya se dijo, es isométricamente isomorfo a un subespacio cerrado de l^∞). Por tanto, si demostramos que $\mathcal{L}$ es una contracción, el teorema del punto de fijo de Banach nos permite establecer que un solo V es punto fijo de $\mathcal{L}$.

Para ver que $\mathcal{L}$ es contracción, tomamos $U, V \in \mathcal{V}$. Fijamos $s \in S$ y suponemos, sin pérdida de generalidad, que $\mathcal{L}V(s) \geq \mathcal{L}U(s)$. Tomamos un $\epsilon > 0$ cualquiera y escogemos $q \in D^{MD}$ de tal modo que $r_q(s) + \lambda(P_q V)(s) \geq \mathcal{L}V(s) - \epsilon$. Es fácil ver

ahora que

$$\begin{aligned}0 \leq &\mathcal{L}V(s) - \mathcal{L}U(s) \leq r_q(s) + \lambda(P_qV)(s) + \epsilon - r_q(s) - \lambda(P_qU)(s) = \\ &= \lambda(P_qV)(s) - \lambda(P_qU)(s) + \epsilon \leq \lambda\|V - U\|_{\mathcal{V}} + \epsilon.\end{aligned}$$

Como ϵ es arbitrario y s es un estado cualquiera de S, hemos terminado.

□

A partir de este hecho se puede expresar un resultado que, en cierto modo, ya había sido deducido en la demostración del Teorema 2.3.15. Puesto que luego vamos a echar mano de él, lo enunciamos como corolario.

[2.3.17] Corolario. *Denotamos por $\pi = q^\infty$ a la política estacionaria $\pi \in \Pi^{MR}$ tal que $\pi = (q, q, ...)$. Entonces, se tiene que $V_\lambda^{q^\infty}$ es la única solución de*

$$V_\lambda^{q^\infty}(s) = r_q(s) + \lambda \sum_{j \in S} P_q(j|s) V_\lambda^{q^\infty}(j).$$

[2.3.17] *Demostración.* Fijándonos los Teoremas 2.3.15 y 2.3.16, observemos que si en vez de tomar el conjunto D^{MR} solo consideramos $D = \{q\}$ se cumple que

- $V_\lambda^* = \sup_{\pi \in D}\{V_\lambda^\pi\} = V_\lambda^{q^\infty}$;
- $\mathcal{L}V = \sup_{q \in D}\{r_q(s) + \lambda \sum_{j \in S} P_q(j|s)V(j)\} = r_q(s) + \lambda \sum_{j \in S} P_q(j|s)V(j)$.

Se puede comprobar que lo desarrollado anteriormente aplica a esta situación, por lo que este operador de Bellman tiene un solo punto fijo que además ha de coincidir con $V_\lambda^* = V_\lambda^{q^\infty}$.

□

Recapitulando, podemos concluir que se ha alcanzado el objetivo que perseguía esta subsección: hemos encontrado una ecuación para la que hemos demostrado que V_λ^* es solución única. Además, dicha ecuación presenta una estructura de punto fijo que nos sugiere un método sencillo para aproximar su solución numéricamente. En la siguiente subsección, vemos como utilizar esto para el diseño de políticas arbitrariamente cerca de las óptimas.

2.3.2. Políticas ϵ-óptimas y su cálculo

Como ya se comentó, decimos que una política π^* es óptima si satisface que $V_\lambda^* = V_\lambda^{\pi^*}$. En general, no se dispone de ninguna garantía de que una política tal

exista, y de hecho no es difícil encontrar ejemplos sencillos (ver [31]) en los no es posible hablar de esa π^*. Ante esta situación, es posible realizar un discusión detallada demostrando la suficiencia de ciertas condiciones para evitar esta patología. No obstante, teniendo en mente las aplicaciones prácticas, se trata de una cuestión soslayable si centramos nuestra aplicación en las políticas ϵ-óptimas.

[2.3.18] Definición. *Decimos que una política es ϵ-óptima (y la denotamos por π^*_ϵ) si satisface que para $\epsilon > 0$*

$$V_\lambda^{\pi^*_\epsilon} \geq V_\lambda^* - \epsilon.$$

Afortunadamente, como nos muestra el siguiente teorema, las políticas ϵ-óptimas se pueden obtener fijándonos en la clase de políticas más sencillas: las políticas deterministas estacionarias de Markov. Recordamos que éstas son del tipo $\pi = (q, q, ...)$ con $q(\cdot|s)$ concentrado en un elemento $a_q(s) \in A_s$ y, en pos de la brevedad, recuperamos la notación $\pi = (q, q, ...) = q^\infty$.

[2.3.19] Teorema. *Para todo $\epsilon > 0$ existe una política estacionaria determinista de Markov ϵ-óptima.*

[2.3.19] *Demostración.* Por el Teorema 2.3.16, $\mathcal{L}V_\lambda^* = V_\lambda^*$. Es posible entonces tomar un $q_\epsilon \in D^{MD}$ tal que

$$r_{q_\epsilon} + \lambda P_{q_\epsilon} V_\lambda^* \geq \sup_{q \in D^{MD}} \{r_q + \lambda P_q V_\lambda^*\} - (1-\lambda)\epsilon e = V_\lambda^* - (1-\lambda)\epsilon e. \tag{2.7}$$

Por el Corolario 2.3.17, tenemos que $V_\lambda^{q_\epsilon^\infty} = (I - \lambda P_{q_\epsilon})^{-1} r_{q_\epsilon}$. Por otro lado, tal y como se discutió en la prueba del Teorema 2.3.15, operador $(I - \lambda P_{q_\epsilon})^{-1}$ es positivo. De este modo, podemos reorganizar (2.7) del siguiente modo,

$$r_{q_\epsilon} - (I - \lambda P_{q_\epsilon}) V_\lambda^* + (1-\lambda)\epsilon e \geq 0,$$

y aplicar a ambos lados el operador $(I - \lambda P_q)^{-1}$ para concluir.

□

Nos encontramos ya en situación de abordar con rigor la idea de la que partimos. Recordamos que el método que sugerimos en principio consistía en encontrar una aproximación V_λ^n de V_λ^* y encontrar la primera ley de decisión intentando maximizar $r_q + \lambda P_q V_\lambda^n$. Este razonamiento, junto con la afirmación del teorema anterior, nos invita a pensar que la política estacionaria $\pi = q^\infty$ puede dar lugar a una $V_\lambda^{q^\infty}$ adecuadamente cerca de V_λ^*.

La formalización de este procedimiento es lo que se conoce como *Value iteration*, debido a que el método de aproximación de V_λ^* se basa en las conclusiones del Teorema 2.3.16. A continuación detallamos el algoritmo y posteriormente mostramos que converge adecuadamente.

Value iteration para políticas $(\epsilon+\delta)$-óptimas

1. Tomamos un $V_\lambda^0 \in \mathcal{V}$ cualquiera, fijamos $\epsilon > 0$ y $\delta > 0$ y declaramos $n = 0$;

2. Para cada $s \in S$, calculamos $V_\lambda^{n+1}(s)$ de acuerdo

$$V_\lambda^{n+1}(s) = \sum_{j\in S} r(s, a_n(s), j)p(j|s, a) + \lambda \sum_{j\in S} p(j|s, a_n(s))V_\lambda^n(j),$$

   donde $a_n : S \to A$ con $a_n(s) \in A_s$ es una función cualquiera que garantiza que se satisfaga

$$V_\lambda^{n+1}(s) \geq \sup_{a\in A_s}\{\sum_{j\in S} r(s, a, j)p(j|s, a) + \lambda \sum_{j\in S} p(j|s, a)V_\lambda^n(j)\} - \frac{1-\lambda}{3}\delta;$$

3. Si $\|V_\lambda^{n+1} - V_\lambda^n\|_{\mathcal{V}} < \frac{1-\lambda}{2\lambda}\epsilon + \frac{\delta}{3}$ pasamos al Paso 4; si no, repetimos el Paso 2;

4. Fijamos $q_{\epsilon+\delta}$ de tal modo que

$$\sum_{j\in S} r(s, a_{q_{\epsilon+\delta}}(s), j)p(j|s, a) + \lambda \sum_{j\in S} p(j|s, a_{q_{\epsilon+\delta}}(s))V_\lambda^{n+1}(j)$$

   sea mayor o igual que

$$\sup_{a\in A_s}\{\sum_{j\in S} r(s, a, j)p(j|s, a) + \lambda \sum_{j\in S} p(j|s, a)V_\lambda^{n+1}(j)\} - \frac{1-\lambda}{3}\delta$$

   y paramos.

Comentario: Obsérvese que si el supremo se puede alcanzar en los Pasos 2 y 4, la posibilidad más natural es implementar el algoritmo tomando las funciones a_n y $a_{q_{\epsilon+\delta}}$ de tal modo que se alcance el máximo. En ese caso, es fácil ver que se pueden repetir los argumentos de la demostración de convergencia con $\delta = 0$ sin que ninguno falle; por tanto, la política asi obtenida satisfacería que

$$V_\lambda^{q^\infty} \geq V_\lambda^* - \epsilon_0 e - \delta e = V_\lambda^* - \epsilon_0 e$$

siendo, pues, ϵ-óptima. Esta situación se da en la mayoría de los casos que se traten computacionalmente, donde se suelen tomar $|A_s| < \infty$.

[2.3.20] Teorema. *El algoritmo de Value iteration finaliza en un número finito de pasos, y la función V_λ^{n+1} en el instante de parada y la política $q_{\epsilon+\delta}^\infty$ obtenidas satisfacen*

1. $\|V_\lambda^{n+1} - V_\lambda^n\|_\mathcal{V} < \frac{\epsilon}{2} + \frac{\delta}{3}$;
2. *la política estacionaria $q_{\epsilon+\delta}^\infty$ es $(\epsilon+\delta)$-óptima.*

[2.3.20] *Demostración.* Comenzamos la demostración argumentando la primera de las afirmaciones del teorema: que la condición de parada se alcanza en un número finito de pasos. Para ello, introducimos en primer lugar la notación

$$\mathcal{L}_{a_q} V(s) = \sum_{j\in S} r(s, a_q(s), j) p(j|s, a_q(s)) + \lambda \sum_{j\in S} p(j|s, a_q(s)) V(j),$$

y la utilizamos para notar que las iteraciones del algoritmo satisfacen

$$\mathcal{L}V_\lambda^n \geq V_\lambda^{n+1} = \mathcal{L}_{a_n} V_\lambda^n \geq \mathcal{L}V_\lambda^n - \frac{1-\lambda}{3}\delta e. \tag{2.8}$$

De este modo, si $u \in \mathcal{V}$ es un vector tal que $0 \leq u \leq 1$, se tiene que

$$V_\lambda^{n+1} = \mathcal{L}V_\lambda^n - \frac{1-\lambda}{3}\delta u. \tag{2.9}$$

Veamos ahora que la distancia entre funciones V_λ^n sucesivas disminuye de acuerdo a un factor constante. Para ello, partimos de que

$$\|V_\lambda^{n+1} - V_\lambda^n\|_\mathcal{V} = \|\mathcal{L}_{a_n} V_\lambda^n - \mathcal{L}_{a_{n-1}} V_\lambda^{n-1}\|_\mathcal{V},$$

y nos fijamos en el tamaño de la diferencia término a término. Si suponemos, por ejemplo, que para un cierto $s \in S$ se tiene que

$$\mathcal{L}_{a_n} V_\lambda^n(s) - \mathcal{L}_{a_{n-1}} V_\lambda^{n-1}(s) \geq 0$$

entonces utilizando (2.8) observaríamos que

$$\begin{aligned} 0 \leq \mathcal{L}_{a_n} V_\lambda^n(s) - \mathcal{L}_{a_{n-1}} V_\lambda^{n-1}(s) \leq \\ \leq \mathcal{L}V_\lambda^n(s) - \mathcal{L}V_\lambda^{n-1}(s) + \frac{1-\lambda}{3}\delta \leq \\ \leq |\mathcal{L}V_\lambda^n(s) - \mathcal{L}V_\lambda^{n-1}(s)| + \frac{1-\lambda}{3}\delta. \end{aligned}$$

Por tanto, razonando análogamente para el caso en que

$$\mathcal{L}_{a_{n-1}} V_\lambda^{n-1}(s) - \mathcal{L}_{a_n} V_\lambda^n(s) \geq 0$$

hemos comprobado que

$$\begin{aligned} \|V_\lambda^{n+1} - V_\lambda^n\|_\mathcal{V} = \|\mathcal{L}_{a_n} V_\lambda^n - \mathcal{L}_{a_{n-1}} V_\lambda^{n-1}\|_\mathcal{V} \leq \\ \leq \|\mathcal{L}V_\lambda^n - \mathcal{L}V_\lambda^{n-1}\|_\mathcal{V} + \frac{1-\lambda}{3}\delta \leq \\ \leq \lambda \|V_\lambda^n - V_\lambda^{n-1}\|_\mathcal{V} + \frac{1-\lambda}{3}\delta. \end{aligned}$$

Una sencilla inducción nos demuestra que podemos acotar ese incremento mediante

$$\|V_\lambda^{n+1} - V_\lambda^n\|_\mathcal{V} \leq \lambda^k \|V_\lambda^{n+1-k} - V_\lambda^{n-k}\|_\mathcal{V} + \frac{1-\lambda}{3}\delta\left(\sum_{i=0}^{k-1}\lambda^i\right).$$

Por tanto, podemos concluir que la parada se alcanza si n es suficientemente grande pues

$$\|V_\lambda^{n+1} - V_\lambda^n\|_\mathcal{V} \leq \lambda^n \|V_\lambda^1 - V_\lambda^0\|_\mathcal{V} + \frac{\delta}{3}.$$

Establecido esto, la primera afirmación es trivial, pues al ser V_λ^* punto fijo de $\mathcal{L}$,

$$\|V_\lambda^* - V_\lambda^{n+1}\|_\mathcal{V} = \|\mathcal{L}V_\lambda^* - \mathcal{L}V_\lambda^n + \frac{1-\lambda}{3}\delta u\|_\mathcal{V} \leq \|\mathcal{L}V_\lambda^* - \mathcal{L}V_\lambda^n\|_\mathcal{V} + \frac{1-\lambda}{3}\delta.$$

Usando que $\mathcal{L}$ es contracción y por la desigualdad triangular, vemos que

$$\|\mathcal{L}V_\lambda^* - \mathcal{L}V_\lambda^n\|_\mathcal{V} \leq \lambda\|V_\lambda^* - V_\lambda^n\|_\mathcal{V} \leq \lambda\|V_\lambda^* - V_\lambda^{n+1}\|_\mathcal{V} + \lambda\|V_\lambda^{n+1} - V_\lambda^n\|_\mathcal{V}.$$

Introduciendo esta desigualdad en la anterior y reordenando se llega al resultado.

Para ver la segunda afirmación, recordamos que - tal y como nos confirmaba el Corolario 2.3.17 - $\mathcal{L}_{a_{q_{\epsilon+\delta}}}$ es contractivo y se cumple que $V_\lambda^{q_{\epsilon+\delta}^\infty} = \mathcal{L}_{a_{q_{\epsilon+\delta}}} V_\lambda^{q_{\epsilon+\delta}^\infty}$.

Por otro lado, se tiene que, por las condiciones del Paso 4,

$$\mathcal{L}V_\lambda^{n+1} \geq \mathcal{L}_{q_{\epsilon+\delta}} V_\lambda^{n+1} \geq \mathcal{L}V_\lambda^{n+1} - \frac{1-\lambda}{3}\delta e.$$

Por tanto, siendo $u \in \mathcal{V}$ un vector tal que $0 \leq u \leq 1$, se cumple que

$$\mathcal{L}_{a_{q_{\epsilon+\delta}}} V_\lambda^{n+1} = \mathcal{L}V_\lambda^{n+1} - \frac{1-\lambda}{3}\delta u.$$

A partir de aquí, notamos que

$$\begin{aligned}
\|V_\lambda^{q_{\epsilon+\delta}^\infty} - V_\lambda^{n+1}\|_\mathcal{V} &= \|\mathcal{L}_{a_{q_{\epsilon+\delta}}} V_\lambda^{q_{\epsilon+\delta}^\infty} - V_\lambda^{n+1}\|_\mathcal{V} \leq \\
&\leq \|\mathcal{L}_{a_{q_{\epsilon+\delta}}} V_\lambda^{q_{\epsilon+\delta}^\infty} - \mathcal{L}_{a_{q_{\epsilon+\delta}}} V_\lambda^{n+1}\|_\mathcal{V} + \|\mathcal{L}_{a_{q_{\epsilon+\delta}}} V_\lambda^{n+1} - V_\lambda^{n+1}\|_\mathcal{V} \leq \\
&\leq \lambda\|V_\lambda^{q_{\epsilon+\delta}^\infty} - V_\lambda^{n+1}\|_\mathcal{V} + \|\mathcal{L}_{a_{q_{\epsilon+\delta}}} V_\lambda^{n+1} - \mathcal{L}V_\lambda^{n+1}\|_\mathcal{V} + \|\mathcal{L}V_\lambda^{n+1} - V_\lambda^{n+1}\|_\mathcal{V} \leq \\
&\leq \lambda\|V_\lambda^{q_{\epsilon+\delta}^\infty} - V_\lambda^{n+1}\|_\mathcal{V} + \frac{1-\lambda}{3}\delta + \lambda\|V_\lambda^{n+1} - V_\lambda^n\|_\mathcal{V} + \frac{1-\lambda}{3}\delta,
\end{aligned}$$

donde en la última desigualdad hemos utilizado (2.9). Reordenando, vemos que $\|V_\lambda^{q_{\epsilon+\delta}^\infty} - V_\lambda^{n+1}\|_\mathcal{V} \leq \frac{\epsilon}{2} + \frac{2}{3}\delta$.

Para terminar, basta notar que

$$\|V_\lambda^* - V_\lambda^{q_{\epsilon+\delta}^\infty}\|_\mathcal{V} \leq \|V_\lambda^* - V_\lambda^{n+1}\|_\mathcal{V} + \|V_\lambda^{q_{\epsilon+\delta}^\infty} - V_\lambda^{n+1}\|_\mathcal{V} = \epsilon + \delta.$$

□

Comentario: La iteración de la función valor no es la única posibilidad para aproximar la política óptima para procesos de decisión de Markov o modelos subyacentes similares. Otras técnicas habituales son la iteración de la política (ver [31]) o la utilización del marco de trabajo de la programación lineal (ver [11] o [25]). En esta memoria hemos escogido presentar el método de la iteración de la función valor por tratarse del enfoque más *universal* de los tres en la teoría de control óptimo.

Comentario: Es fácil observar que el algoritmo de la iteración de la función valor no se puede simular numéricamente de manera directa si $|S|$ o $|A|$ son infinitos. Este tipo de situaciones de abordan encontrando en primer lugar una familia de procesos de decisión de Markov finitos que aproximan el proceso original. A continuación, se estima el error asociado a resolver el problema aproximado y el original; a partir de ahí, cuando se tenga garantía de que ese error se encuentra dentro de una tolerancia determinada previamente, se resuelve numéricamente el problema finito usando - por ejemplo - el algoritmo de esta sección. El lector interesado es remitido a [31].

Aunque se podrían discutir ahora técnicas para estimar y acelerar la convergencia de estos algoritmos, el problema de control óptimo para los procesos de decisión de Markov con el criterio de la recompensa total descontada esperada ya está conceptualmente resuelto. En lugar de eso, consideramos preferible incluir una pequeña ilustración de lo discutido hasta ahora mediante un ejemplo numérico y de ahí pasar a estudiar en el próximo capítulo cómo estimar la política óptima de un modelo desconocido a partir de observaciones.

2.4. Aplicación numérica

Con el objeto de ilustrar el tipo de problemas que se puede abordar con el método de *Value iteration*, introducimos en esta sección dos problemas sencillos. Como se ha podido comprobar, la iteración de la función valor es un procedimiento teóricamente sólido y ampliamente aceptado y testado; por tanto, la idea será utilizar los resultados que presentamos a continuación como referencia para el algoritmo que estudiaremos en el próximo capítulo.

El primero de los problemas que consideramos es el de un agente que busca salir de una habitación con la máxima agilidad posible evitando el impacto contra los muros. El agente debe decir en cada instante si quiere moverse hacia arriba, abajo,

derecha, izquierda o quedarse quieto, pero sobre su acción se solapará la presencia de un ruido que lo empuja de manera aleatoria. En la salida se encuentran unos estados absorbentes que simbolizan la finalización del proceso.

Para tratar esta situación usando el lenguaje de los procesos de decisión de Markov, consideramos el espacio de estados $S = \{1, 2, ..., N\}^2$ con N impar, donde, por ejemplo, la esquina superior izquierda sería el valor $(1, 1)$ y la inferior derecha (N, N). De entre estos estados, el subconjunto

$$E = \left\{(s_1, s_2) \in S : 1 \leq s_1 \leq 2, \left\lfloor \frac{N}{2} \right\rfloor + 1 - 2 \leq s_2 \leq \left\lfloor \frac{N}{2} \right\rfloor + 1 + 2\right\}$$

representa la salida. El conjunto de acciones sería ahora $A = \{u_p, d_o, r_i, l_e, q_u, n_c\}$, donde la acción n_c sirve para modelar estados s en los que no ha habido elección posible (por ejemplo, los estado absorbentes que se alcanzan al llegar a la puerta y que representan el fin de la tarea). Por su parte, las transiciones vienen regidas por expresiones como la siguiente,

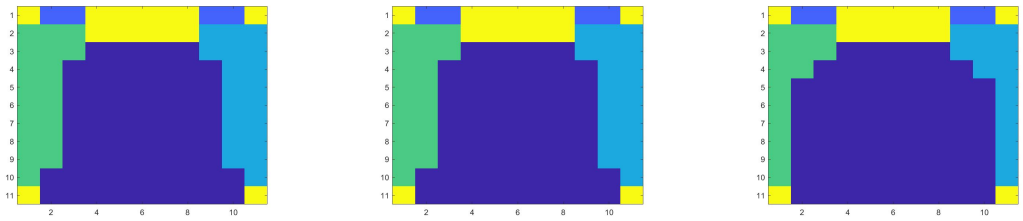
$$s_{t+1} = (s_{1;t+1}, s_{2;t+1}) = f(s_t, u_p, w_t) = (s_{1;t}, s_{2;t}) + (-1, 0) + w_t,$$

con los correspondientes ajustes para lidiar con las particularidades de bordes y esquinas. En la expresión anterior, las $\{w_t\}_{t\in\mathbb{N}_0}$ son variables aleatorias independientes uniformemente distribuidas entre los valores $(0, 1)$,$(0, -1)$, $(1, 0)$ y $(-1, 0)$. Finalmente, la función $r : S \times A \times S \to \mathbb{R}$ toma la forma $r(s, a, j) = \tilde{r}(s)$ y vale 1 si $s \in E$ y $-0{,}1$ si s representa un borde de la habitación.

Comentario: El presente problema captura adecuadamente el fenómeno para el que se ideó originalmente el principio de optimalidad de Bellman. Obsérvese que centrándonos meramente en la recompensa *local* asociada a una acción no es posible evaluar su calidad; es necesario, pues, considerar, las recompensas futuras o que *se reciben con retraso*. De este fenómeno surge el nombre del influyente trabajo *Learning from delayed rewards* ([41]), que desarrollaremos en el próximo capítulo.

Para distintos valores de λ (0,5, 0,7 y 0,9) corremos el algoritmo con $V_0 \equiv 0$ y $\epsilon = 10^{-6}$ y obtenemos las políticas estacionarias asociadas que se pueden observar en la Figura 2.2. Obsérvese que la recompensa futura para valores más bajos de λ es más pequeña, por lo que se prioriza más el evitar la penalización presente que se recibe si se tocan los bordes.

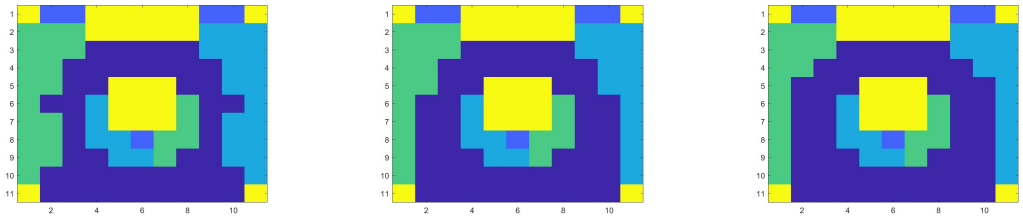
Por otro lado, con el objeto de ver cómo responde tanto este algoritmo como el que introduciremos en el Capítulo 3, es posible incluir obstáculos que introduzcan la necesidad de considerar caminos no tan sencillos. Obsérvese que, dado que la evolución de nuestro sistema está sometida a un ruido del orden del efecto de las propias acciones, no tiene sentido exigir mucha complejidad y precisión en el *laberinto* resultante. Por tanto, y a modo de ejemplo, trabajaremos con un solo obstáculo ubicado en el centro



(a) $\lambda = 0{,}5$ (b) $\lambda = 0{,}7$ (c) $\lambda = 0{,}9$

Figura 2.2: Políticas estacionarias obtenidas para distintos valores de λ. Código de colores: azul oscuro → arriba, azul → abajo, verde → derecha, azul claro → izquierda, amarillo → no hay elección.

de la habitación, que introducimos matemáticamente mediante la consideración de unos estados absorbentes con recompensa -1. Las nuevas políticas estacionarias que sugiere el algoritmo con $V_0 = 0$ y $\epsilon = 10^{-6}$ se ilustran en la Figura 2.3.



(a) $\lambda = 0{,}5$ (b) $\lambda = 0{,}7$ (c) $\lambda = 0{,}9$

Figura 2.3: Políticas estacionarias obtenidas para distintos valores de λ. Código de colores: azul oscuro → arriba, azul → abajo, verde → derecha, azul claro → izquierda, amarillo → no hay elección.

Cabe destacar, eso sí, que se puede comprobar que el algoritmo efectivamente converge de manera adecuada en habitaciones más grandes que permiten introducir *laberintos* más complicados. Así, las limitaciones solo comienzan a surgir cuando el número de pasos necesarios para llegar a la salida es tan grande que, por el efecto del descuento, la función valor toma valores que se confunden con la precisión de la máquina.

Con el objeto de comprobar que algunas de las conclusiones que extraeremos más tarde no se deben a factores idiosincráticos del proceso de decisión de Markov recién descrito, introducimos un segundo ejemplo de problema que se puede abordar con las técnicas introducidas en este capítulo. Para ello, vamos a considerar otro proceso que busca describir la situación de un agente cuyo objetivo es esquivar el impacto de unos *proyectiles* mediante movimientos horizontales. Los proyectiles caen desde arriba hacia abajo y en su trayectoria sufren perturbaciones que los empujan hacia la derecha

y la izquierda con probabilidad $\frac{1}{2}$.

(a) Instante 1 (b) Instante 2 (c) Instante 3 (d) Instante 4

Figura 2.4: Instantes sucesivos del proceso de decisión de Markov. Las casillas en color cian indican la posición de los proyectiles, y la amarilla señala la posición del agente.

En cada instante de tiempo, como se esquematiza en la Figura 2.4, el proyectil que se encontraba en la fila superior en la columna j cae en la fila inferior a la columna $j+1$ o $j-1$ (si $j=1$, caerá a $j=1$ o $j=2$; por otro lado, si $j=N$, caerá a $j=N-1$ o $j=N$). Con el fin de evitar ese impacto, el agente se puede mover a la derecha, a la izquierda o quedarse quieto. Además, en la fila superior aparece un nuevo proyectil con una distribución uniforme. Con el objeto de encapsular matemáticamente la exigencia de que el agente esquive esos golpes, éste recibirá $+1$ de recompensa si no es alcanzado, mientras que si es impactado recibirá -1.

La escritura matemática de este proceso se lleva a cabo tomando $S=\{1,2,...,N\}^3$, donde la primera coordenada indica la columna del proyectil de la fila superior, la segunda indica la columna del proyectil de la fila inferior y la tercera la columna en la que se encuentra el agente; vemos, pues, que incluso problemas tan sencillos como este pueden presentar una cantidad de estados $|S|=N^3$ significativa. Por su parte, el conjunto de acciones A se reduce a $\{r_i, q_u, l_e\}$ (que representan, respectivamente, derecha, quieto e izquierda).

No es difícil pensar en múltiples políticas deterministas estacionarias óptimas para esta situación si observamos que el agente siempre puede evitar el impacto si no se encuentra en un extremo y, efectivamente, aplicando el algoritmo de iteración de la función valor con un ϵ suficientemente pequeño es fácil demostrar que se alcanza alguna de éstas. La tridimensionalidad del espacio de estados complica la visualización; por este motivo, incluiremos meramente la acción que sugiere en ciertas ocasiones la política obtenida usando el mencionado algoritmo con condición inicial $V_0 \equiv 0$, $\epsilon = 10^{-6}$ y $\lambda = \frac{1}{2}$ (ver Figura 2.5).

En lo que refiere al ritmo de la convergencia del algoritmo de la iteración de valor para estos problemas, no nos detendremos aquí tampoco, pues argumentando como es habitual en iteraciones de punto fijo es sencillo comprobar tanto analítica como experimentalmente que el error en la norma $\|\cdot\|_{\mathcal{V}}$ de la aproximación V_n es $\mathcal{O}(\lambda^n)$, y

(a) Acción q_u (b) Acción l_e (c) Acción: q_u (d) Acción: r_i

Figura 2.5: Ejemplo de distintos instantes (no consecutivos) del proceso de decisión de Markov. En el pie de la imagen se indica la acción que sugiere la política obtenida mediante el método de iteración de valor.

que el número de iteraciones necesario es

$$\frac{\log\left(\frac{\epsilon(1-\lambda)}{2C\lambda}\right)}{\log\lambda}.$$

Como ya se indicó, el lector interesado en la aceleración de la convergencia de este algoritmo es remitido a [31].

CAPÍTULO 3
Q-learning

En el capítulo anterior se ha definido y estudiado las propiedades de un modelo probabilístico que determina cómo evoluciona el estado de un agente de acuerdo a las decisiones que toma. El de los procesos de decisión de Markov es un modelo muy general, en el que es posible captar muchas dinámicas diferentes (desde complejas transiciones estocásticas hasta el caso degenerado determinista); a cambio, el usuario ha de especificar una gran cantidad de grados de libertad. En el caso tratado en el Capítulo 2, el número de parámetros a establecer es del orden de $|S|^2|A|$ (los núcleos de Markov $p(j|s,a)$ vienen dados por $\sum_{s\in S}|S||A_s|$ números, al igual que la función de recompensa $r(s,a,j)$).

En muchas situaciones prácticas, nos topamos con el problema abstracto de encontrar una ley de control *óptima* para la dinámica de un agente que recibe unas recompensas cuya estructura es sabida pero que está sometido a unas reglas de evolución desconocidas. Los procesos de decisión de Markov se presentan en este caso como un candidato natural para llevar a cabo el papel de modelo subyacente, trasladando así la dificultad a realizar estimaciones estadísticas de los núcleos de Markov $p(j|s,a)$ con el objeto de, posteriormente, encontrar una política siguiendo los métodos del Capítulo 2.

No obstante, en el Capítulo 1 ya se introdujeron las dos cuestiones que esta manera de proceder plantea. En primer lugar, está la pregunta de la eficacia, relacionada con el interrogante que surge sobre el empalme adecuado entre la estimación y la optimización: ¿convergerán las políticas ϵ-óptimas de los modelos de Markov estimados a una política ϵ-óptima del modelo de Markov *verdadero*? En segundo lugar, cabe preguntarse sobre la eficiencia de este enfoque: si lo que en el fondo queremos conocer es la política óptima, ¿por qué no intentar estimarla de una manera más directa y ahorrar cálculos asociados a aproximaciones del modelo que solo nos interesan indirectamente? En los próximos párrafos, presentamos el método de Q-learning (introducido en [41]), que consigue soslayar ambas cuestiones estimando meramente una magnitud auxiliar - la función Q - a partir de la cuál se puede (bajo circunstancias relativamente generales) sintetizar políticas óptimas.

El objetivo de este capítulo es demostrar que la estimación de Q-learning efectivamente converge cuando todos los pares (s, a) de estado-acción son visitados infinitas veces, así como discutir el problema de la exploración. El objeto de éste último es preguntarse cómo organizar esas visitas, y es de importancia cuando se trata con sistemas con un número muy grande de estados y acciones (de tal modo que pueda ser excesivamente costoso visitarlos todos con la misma frecuencia) o cuando el desconocimiento del sistema subyacente es tal que ni siquiera se dispone de una descripción completa sobre cuáles son los posibles pares (s, a).

Terminamos esta introducción comentando en primer lugar que, debido al enfoque eminentemente computacional de los métodos que se van a presentar en este capítulo, resulta natural restringirse al caso en el que $|S|$ y $|A|$ son finitos; nótese que, de ser $|S|$ o $|A|$ infinito, sería imposible visitar al menos una vez todos los pares (s, a) en un número de observaciones finito. Por otro lado, aprovechamos este último párrafo para aclarar que, aunque aquí se va a presentar para el problema de la recompensa total descontada esperada, la validez y aplicaciones del método Q-learning no se reducen simplemente a este caso concreto; las ideas esenciales pueden ser adaptadas para abordar también otras familias de problemas (ver [4]).

3.1. Sobre el método y su convergencia

El marco de trabajo es el siguiente: suponemos que un agente evoluciona de acuerdo a un proceso de Markov $\{\mathbb{N}_0, S, \{A_s\}_{s\in S}, p(j|s,a), r(s,a,j)\}$ con $|S| < \infty$ y $|A_s| < \infty$. Sobre ese modelo no conocemos ninguno de los valores de $p(\cdot|\cdot,\cdot)$, pero podemos realizar observaciones (ya sea mediante simulaciones o utilizando datos almacenados) sobre el resultado de ejecutar la acción a en el estado s. Nuestro objetivo es desarrollar, a partir de esa información, un método para diseñar una política lo más adecuada posible de acuerdo al criterio de la esperanza total descontada esperada.

Lo primero que observamos es que, como consecuencia del Teorema 2.3.19, en este tipo de modelos finitos existe una política óptima estacionaria determinista.

[3.1.1] Corolario. *Si $|S| < \infty$, $|A_s| < \infty$, entonces es posible encontrar una política óptima determinista de Markov del tipo $\pi^* = q^\infty = (q, q, ...)$, donde la probabilidad $q(\cdot|s)$ está concentrada en un punto al que denominaremos $a_q(s) \in A_s$.*

[3.1.1] *Demostración.* Utilizaremos a partir de ahora la notación $S \times A_S = \{(s,a) \in S \times A : a \in A_s\}$, y $A_S^S = \{a_q \in A^S : (s, a_q(s)) \in S \times A_S \ \forall s \in S\}$. Es claro que existe una identificación entre el conjunto A_S^S y el conjunto Π^{SD} de políticas estacionarias deterministas, y que por tanto el número de posibles políticas estacionarias deterministas es finito (en concreto, $\prod_{s\in S} |A_s|$).

Recordamos ahora que el Teorema 2.3.19 nos decía que siempre es posible encontrar una política estacionaria determinista ϵ-óptima. Si suponemos ahora que en esta situación no hay ninguna política óptima en el conjunto finito Π^{SD}, entonces sería posible encontrar un ϵ que contradijese el Teorema 2.3.19.

□

Por tanto, en lo que resta de capítulo podemos centrarnos en las políticas estacionarias deterministas q^∞ que, de acuerdo a la notación introducida en la demostración anterior, quedan determinadas mediante una función $a_q \in A_S^S$. Así pues, se va a formular nuestro método de tal modo que simplemente nos devuelva en cada s un $a_q(s)$.

3.1.1. Motivación

Para ver qué condiciones ha de cumplir el mapa a_q para dar lugar a una política óptima, supongamos primero que efectivamente $q^\infty = \pi^*$. Se tiene entonces que $V_\lambda^* = V_\lambda^{q^\infty}$ y que, por el Corolario 2.3.17, se cumple la igualdad $V_\lambda^{q^\infty} = \mathcal{L}_{a_q} V_\lambda^{q^\infty}$. De este modo, observamos que

$$\mathcal{L}_{a_q} V_\lambda^* = \mathcal{L}_{a_q} V_\lambda^{q^\infty} = V_\lambda^{q^\infty} = V_\lambda^* = \mathcal{L} V_\lambda^*.$$

Si en vez de eso comenzamos suponiendo que $\mathcal{L}_{a_q} V_\lambda^* = \mathcal{L} V_\lambda^*$, empezamos notando que $\mathcal{L}_{a_q} V_\lambda^* = \mathcal{L} V_\lambda^* = V_\lambda^*$. Puesto que $V_\lambda^{q^\infty}$ es punto fijo de $\mathcal{L}_{a_q}$ - y éste ha de ser único por ser el operador contractivo - se tiene que $V_\lambda^{q^\infty} = V_\lambda^*$. De este modo, hemos mostrado el siguiente resultado.

[3.1.2] Teorema. *Bajo las hipótesis de este capítulo, una función a_q tiene una política estacionaria asociada óptima si y solo si*

$$\sum_{j\in S} r(s, a_q(s), j) p(j|s, a_q(s)) + \lambda \sum_{j\in S} p(j|s, a_q(s)) V_\lambda^*(j) =$$
$$= \max_{a\in A_s} \{\sum_{j\in S} r(s, a, j) p(j|s, a) + \lambda \sum_{j\in S} p(j|s, a) V_\lambda^*(j)\}.$$

Por tanto, del teorema anterior se concluye que con el objetivo de encontrar la función a_q basta conocer los valores de $\sum_{j\in S} r(s, a, j) p(j|s, a) + \lambda \sum_{j\in S} p(j|s, a) V_\lambda^*(j)$ para todo $(s, a) \in S \times A_S$; resulta lógico, pues, poner esa magnitud en el centro de nuestros esfuerzos de estimación.

[3.1.3] Definición. *Dado un proceso de decisión de Markov, definimos la función $Q^* : S \times A_S \to \mathbb{R}$ mediante*

$$Q^*(s, a) := \sum_{j\in S} r(s, a, j) p(j|s, a) + \lambda \sum_{j\in S} p(j|s, a) V_\lambda^*(j).$$

Comentario: La interpretación de la función Q^* es que en el par (s, a) su valor es la máxima recompensa total descontada esperada si se lleva a cabo la acción a. Es natural, pues, que $a_{q^*}(s) = \arg\max_{a \in A_s} Q^*(s, a)$. Por otro lado, de la ecuación de Bellman se deduce que $V_\lambda^*(s) = \mathcal{L}V_\lambda^*(s) = \max_{a \in A_s} Q^*(s, a)$.

Comentario: Con este *cambio de variable* en nuestro problema de estimación obtenemos un problema en principio más manejable. Obsérvese que pasamos de tener las $\sum_{s \in S} |S||A_s|$ incógnitas a estimar de los núcleos $p(j|s, a)$ (con los que después habría que, además, calcular la función V_λ^*) a tener las $\sum_{s \in S} |A_s|$ incógnitas de la función Q^*.

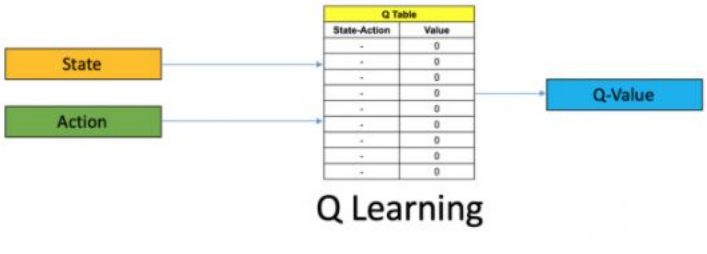


Figura 3.1: A la función $Q^* : S \times A_S \to \mathbb{R}$ a veces se le denomina en la literatura como Q *tabla* por la forma en la que es almacenada. En este trabajo buscaremos aproximar la función Q^* con funciones o *tablas* $\{Q_t\}_{t \in \mathbb{N}_0}$; no obstante, cuando las dimensiones son muy grandes es útil aproximar Q^* utilizando redes neuronales, lo que da lugar al *Deep Q-learning*.

De cara a la obtención de la función V_λ^*, un paso crucial fue estudiar las propiedades que había de satisfacer para encontrar así qué ecuación resolvía. Puesto que la relación entre las funciones V_λ^* y Q^* es tan íntima, cabe esperar que Q^* sea también punto fijo de algún operador. Si nos fijamos en la definición de Q^* y la igualdad $V_\lambda^*(s) = \max_{a \in A_s} Q^*(s, a)$, comenzamos notando que

$$Q^*(s, a) = \sum_{j \in S} p(j|s, a) \left(r(s, a, j) + \lambda \max_{b \in A_j} Q^*(j, b) \right). \tag{3.1}$$

Por tanto, llamando $\mathcal{Q} = \{Q \in \mathbb{R}^{S \times A_S} : \|Q\|_\infty < \infty\}$, se nos está sugiriendo considerar el operador $H : \mathcal{Q} \to \mathcal{Q}$ definido mediante

$$(HQ)(s, a) := \sum_{j \in S} p(j|s, a) \left(r(s, a, j) + \lambda \max_{b \in A_j} Q(j, b) \right).$$

Esta definición es la adecuada, pues da lugar a otro operador contractivo.

[3.1.4] Teorema. *El operador $H : \mathcal{Q} \to \mathcal{Q}$ es contractivo y su único punto fijo es la función Q^*.*

[3.1.4] *Demostración.* El espacio vectorial $\mathcal{Q}$ es de Banach (es isométricamente isomorfo a un subespacio cerrado de l^∞), por lo que si demostramos que H es contractivo podemos concluir que tiene un único punto fijo. De (3.1), concluiríamos que es Q^*.

La contractividad de H es trivial observando primero que

$$|(HQ_1)(s,a) - (HQ_2)(s,a)| = \lambda \left| \sum_{j \in S} p(j|s,a) \left(\max_{b \in A_j} Q_1(j,b) - \max_{b \in A_j} Q_2(j,b) \right) \right|.$$

Fijamos un $j \in S$ y suponiendo que, por ejemplo, $\max_{b \in A_j} Q_2(j,b) - \max_{b \in A_j} Q_1(j,b) \geq 0$, vemos que

$$\begin{aligned} 0 \leq |\max_{b \in A_j} Q_2(j,b) - \max_{b \in A_j} Q_1(j,b)| &= \max_{b \in A_j} Q_2(j,b) - \max_{b \in A_j} Q_1(j,b) \leq \\ &\leq Q_2(j,\tilde{b}) - Q_1(j,\tilde{b}) = |Q_2(j,\tilde{b}) - Q_1(j,\tilde{b})| \leq \|Q_2 - Q_1\|_{\mathcal{Q}}. \end{aligned}$$

De estas dos relaciones se deduce que $|(HQ_1)(s,a) - (HQ_2)(s,a)| \leq \lambda \|Q_1 - Q_2\|_{\mathcal{Q}}$.

□

A diferencia del operador $\mathcal{L}$ en el Capítulo 2, en la situación que ahora abordamos no se puede utilizar de manera directa el operador H (¡recuérdese que desconocemos los valores de $p(j|s,a)$!). No obstante, desempeñará un papel importante de cara a análisis teóricos relacionados con la convergencia de nuestro algoritmo.

¿Qué es lo que si se puede utilizar de cara a definir nuestro método iterativo? Desde luego, habremos de conocer los valores de la función $r : S \times A \times S$ (¿cómo intentar diseñar una política óptima si ni siquiera nos dicen cómo nos dan las recompensas?) y el valor de λ pues viene fijado por el propio criterio que se busca maximizar. Asimismo, en la iteración t de nuestro algoritmo (no se debe confundir las iteraciones t del algoritmo con los instantes temporales del proceso; en principio no tiene por qué haber relación alguna) dispondremos de una aproximación de $Q^*(s,a)$ a la que denominamos $Q_t(s,a)$, junto con la posibilidad de realizar una simulación.

Realizamos ahora el siguiente razonamiento: llamando $s'_t(s,a)$ al estado que nos devuelve la simulación cuando aplicamos la acción a en el instante s, vemos que

$$Q^*(s,a) = (HQ^*)(s,a) = \mathbb{E}\Big[r(s,a,s'_t(s,a)) + \lambda \max_{b \in A_{s'_t(s,a)}} Q^*(s'_t(s,a),b) \Big].$$

Del mismo modo que para estimar una esperanza $\mathbb{E}[X] \approx E_t$ se utiliza la iteración

$$E_{t+1} = \frac{1}{t+1} \sum_{\tau=0}^{t} X_\tau = (1 - \frac{1}{t+1}) E_t + \frac{1}{t+1} X_t, \tag{3.2}$$

si escogemos las simulaciones $s'_t(s,a)$ independientes parece razonable escribir

$$Q_{t+1}(s,a) = (1 - \frac{1}{t+1})Q_t(s,a) + \frac{1}{t+1}\Big[r(s,a,s'_t(s,a)) + \lambda \max_{b \in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b)\Big].$$

Para tratar el problema con mayor generalidad, tomamos $\alpha_t(s,a) \in [0,1]$ y nos basamos en las ideas anteriores para definir la iteración de Q-learning.

[3.1.5] Definición. *Nos referiremos como iteración de Q-learning a, dados* $\alpha_t(s,a) \in [0,1]$*, la igualdad*

$$Q_{t+1}(s,a) = (1-\alpha_t(s,a))Q_t(s,a)+\alpha_t(s,a)\Big[r(s,a,s'_t(s,a))+\lambda \max_{b \in A_{s'_t(s,a)}} Q_t(s'_t(s,a),a)\Big].$$

Comentario: En el caso de la iteración (3.2), es sabido que en muchas circunstancias que tenemos garantizada la convergencia por las Leyes de los Grandes Números. La dificultad que añade la iteración de Q-learning respecto a una situación tan clásica es el hecho de que necesitamos utilizar la propia aproximación al incorporar la información de la observación t. Por tanto, concluir que se tiene convergencia no es tan sencillo como en el caso anterior.

En la siguiente subsección detallamos cómo emplear está iteración y discutimos cómo abordar el algoritmo resultante utilizando la metodología de la teoría de aproximaciones estocásticas.

3.1.2. El algoritmo de Q-learning y el proceso estocástico asociado

Para tratar la iteración de Q-learning de tal modo que se pueda estudiar su convergencia de manera teórica y sea posible dar un algoritmo cerrado para su uso se pueden emplear dos enfoques. Por un lado, se puede considerar la aproximación original de Watkins (ver [41] y [40]); ésta utiliza una idea muy creativa, aunque peca de basarse en conceptos muy particulares y carecer de conexión con otras teorías ya desarrolladas.

Por otro lado, es posible enfocar el problema desde las técnicas y el lenguaje de los métodos de aproximación estocástica. Aunque los teoremas y algoritmos existentes de esta rama de las matemáticas no eran suficientes para abordar con total rigor el algoritmo de Q-learning (ver [39]), los resultados adicionales que se desarrollaron para Q-learning se puedieron encuadrar en un contexto más general. Además, es un marco de trabajo que goza de más flexibilidad que no solo permitió adaptar el Q-learning a otro tipo de situaciones (ver [4]) y establecer su conexión con otros algoritmos existentes (ver [26]) si no que también ayudó a los investigadores a comprender la convergencia en mayor profundidad (ver [19]).

Es éste último el enfoque que desarrollaremos aquí, cuya esencia - de manera resumida - consiste en considerar cada vez que se corre el algoritmo de iteración como una realización de un proceso estocástico. Con el objeto de ilustrar esta idea con más detalle, procedemos en los próximos párrafos a desarrollar el modelo probabilístico canónico para este tipo de circunstancias; no obstante, como es bien sabido, en ningún caso debe deducirse que éste es el único espacio de probabilidad que puede dar lugar al proceso asociado al algoritmo.

Para construir el modelo canónico empezamos considerando una familia de espacios de probabilidad indexados con el conjunto $\mathbb{N}_0$ (estos índices representarán las iteraciones de nuestro algoritmo) que se definen de tal modo que el conjunto Ω_t es el producto cartesiano de $|S \times A_S|$ copias de S. Puesto que Ω_t tiene una cantidad finita de elementos, lo equipamos con la σ-álgebra $\mathcal{P}(\Omega_t)$. La idea de construir este espacio medible es que, para terminar, definimos una probabilidad $\mathbb{P}_t$ utilizando la medida producto de tal modo que las variables aleatorias $s'_t(s,a) : \Omega_t \to S$ sean independientes y sigan una distribución dada por $\mathbb{P}_t(\{s'_t(s,a) = j\}) = p(j|s,a)$. Se puede ver entonces que las variables aleatorias $s'_t(s,a)$ representan el resultado de ejecutar la acción a en el estado s.

A partir de estos espacios de medida, definimos el modelo probabilístico canónico para representar un algoritmo de aproximación estocástica,

$$(\Omega, \mathcal{P}(\Omega), \mathbb{P}) := \bigotimes_{t \in \mathbb{N}_0} (\Omega_t, \mathcal{P}(\Omega_t), \mathbb{P}_t).$$

Al igual que en la construcción del modelo de los procesos de decisión de Markov del Capítulo 2, esta definición de $(\Omega, \mathcal{P}(\Omega), \mathbb{P})$ es válida por el Teorema de Daniell-Kolmogorov ([23]). Sobre este espacio vamos a definir también los vectores aleatorios $s'_t : \Omega \to S \times A_S$ de la manera más directa: si $\pi_t^\Omega : \Omega \to \Omega_t$ es la función que proyecta ω en el espacio Ω_t, definimos - abusando de notación - las $|S \times A_s|$ componentes de s'_t mediante $s'_t(s,a)(\omega) := s'_t(s,a)(\pi_t^\Omega(\omega))$. Introducimos los también vectores aleatorios $\alpha_t : \Omega \to S \times A_S$ de tal modo que sus $|S \times A_S|$ componentes $\alpha_t(s,a)$ satisfagan que $\alpha_t(s,a) \in [0,1]$. Por último, otro vector aleatorio con el que trabajaremos es $r_t : \Omega \to S \times A_S$, cuyas componentes se definen mediante $r_t(s,a)(\omega) = r(s,a,s'_t(s,a)(\omega))$

Estamos también interesados en encapsular matemáticamente la información que va acumulando el algoritmo en un instante $t \in \mathbb{N}_0$, de tal modo que no caigamos en el error de utilizar observaciones aún no disponibles. Es sabido que esto se lleva a cabo introduciendo una filtración, que en nuestro caso toma la siguiente definición: $\{\mathcal{F}_t\}_{t \in \mathbb{N}}$ es la filtración compuesta por las σ-álgebras $\mathcal{F}_t = \sigma(s'_k, k = 1, 2, ..., t-1)$. De este modo se tendrá, entre otras cosas, que r_t es $\mathcal{F}_{t+1}$-medible, y se pueden imponer condiciones como la siguiente:

[3.1.6] Asunción. *Los vectores aleatorios α_t han de ser $\mathcal{F}_t$-medibles.*

Comentario: La idea de la asunción anterior es que, previamente a la observación de los $s'_t(s,a)$, sepamos qué valor hemos de escoger para los $\alpha_t(s,a)$ a la hora de realizar la iteración de Q-learning del instante t. Se pueden considerar otras posibilidades, como por ejemplo permitir cierta aleatoriedad. No obstante, lo que ha de quedar muy claro es que para que se pueda aplicar el algoritmo no se debería permitir que el vector α_t sea función ni del resultado que se obtenga en ese instante $s'_t(s,a)$ ni, desde luego, de resultados que aún no se conocerían (por ejemplo, los vectores $s'_{t+1}, s'_{t+2}, ...$).

Comentario: En muchos casos, la Asunción 3.1.6 se satisface trivialmente puesto que se consideran valores de α_t deterministas. No obstante, esta no es una situación general; por ejemplo, cuando se utiliza algún método de exploración sobre el algoritmo Q-learning, es habitual considerar los valores de $\alpha_t(s,a)$ como una función de las veces que ya se ha visitado el par (s,a) en lo que va de algoritmo. Es fácil ver que estas condiciones satisfacen el marco de trabajo que presentamos aquí.

Tras esta discusión, ya estamos en disposición de definir qué es exactamente desde un punto de vista matemático la iteración de Q-learning.

[3.1.7] Definición. *Sobre el espacio de probabilidad anterior y con la notación discutida en esta subsección, definimos - dado un valor inicial $Q^0 \in \mathcal{Q}$ - las variables aleatorias $Q_0(s,a)(\omega) = Q^0(s,a)$ y*

$$Q_{t+1}(s,a) = (1-\alpha_t(s,a))Q_t(s,a)+\alpha_t(s,a)\Big[r(s,a,s'_t(s,a))+\lambda \max_{b\in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b)\Big].$$

Comentario: Mediante inducción es directo probar que los vectores aleatorios Q_t son $\mathcal{F}_t$-medibles.

De este modo, el marco teórico para abordar nuestro método es concebirlo como una realización de un proceso estocástico: cada vez que corremos el algoritmo, estamos observando en el tiempo los valores del vector $Q_t(\omega)$ para un ω fijo. Decir que el algoritmo converge equivale, pues, a mostrar que el subconjunto de Ω

$$\mathcal{W} = \{\omega \in \Omega : \lim_{t\to\infty} \|Q_t(\omega) - Q^*\|_{\mathcal{Q}} = 0\}$$

tiene probabilidad 1. La prueba de esta afirmación es la tarea que abordamos en la siguiente sección.

3.2. Convergencia de aproximaciones estocásticas

Nuestro objetivo es mostrar la convergencia casi seguro de la iteración de la Definición 3.1.7, y para ello comenzamos notando que la fuente de aleatoriedad en cada

paso del algoritmo viene dada por el término

$$r(s,a,s'_t(s,a)) + \lambda \max_{b \in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b).$$

Con el objetivo de comprender adecuadamente la *dirección* y *magnitud* de la actualización que en media nos introduce cada iteración, es habitual intentar descomponer la componente estocástica de la definición de Q_{t+1} en una parte previsible y un término de media 0. Así pues, pasamos a estudiar

$$\mathbb{E}\left[r(s,a,s'_t(s,a)) + \lambda \max_{b \in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b) \Big| \mathcal{F}_t \right]$$

y vemos que, por la independencia de los vectores $\{s'_k\}_{k\in\mathbb{Z}}$, se cumple que

$$\begin{aligned}
&\mathbb{E}\left[r(s,a,s'_t(s,a)) + \lambda \max_{b \in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b) \Big| \mathcal{F}_t \right] = \\
&\mathbb{E}\left[r(s,a,s'_t(s,a)) + \lambda \max_{b \in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b) \right] = \\
&= \sum_{j\in S} \mathbb{P}(\{s'_t(s,a)=j\}) \left[r(s,a,j) + \lambda \max_{b\in A_j} Q_t(j,b) \right] = \\
&= \sum_{j\in S} p(j|s,a) \left[r(s,a,j) + \lambda \max_{b\in A_j} Q_t(j,b) \right] = (HQ_t)(s,a).
\end{aligned}$$

De este modo, si definimos el vector aleatorio $w_t(s,a)$ de acuerdo a

$$\begin{aligned}
w_t(s,a) = r(s,a,s'_t(s,a)) + \lambda \max_{b \in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b) - \\
- \sum_{j\in S} p(j|s,a) \left[r(s,a,j) + \lambda \max_{b\in A_j} Q_t(j,b) \right].
\end{aligned}$$

observamos (sumando y restando la misma cantidad) que la Definición 3.1.7 es equivalente a la iteración

$$Q_{t+1}(s,a) = (1-\alpha_t(s,a))Q_t(s,a) + \alpha_t(s,a)\Big[(HQ_t)(s,a) + w_t(s,a)\Big]. \tag{3.3}$$

Comentario: Obsérvese que, claramente, $w_t(s,a)$ es $\mathcal{F}_{t+1}$-medible pero en general no lo es para $\mathcal{F}_t$. De hecho, por definición, el interés de $w_t(s,a)$ radica en la *parte no $\mathcal{F}_t$-medible*, pues cumple que $\mathbb{E}\left[w_t(s,a)|\mathcal{F}_t\right] = 0$.

Por otro lado, otro preliminar habitual en el estudio de aproximaciones estocásticas (ya sea dentro o fuera de las propias demostraciones) es centrar el problema y reducirlo a estudiar el decaimiento del error a 0. En nuestro caso, eso equivale a introducir el

vector aleatorio de componentes $E_t(s,a) = Q_t(s,a) - Q^*(s,a)$ y observar, restando Q^* en (3.3), que sigue una dinámica del tipo

$$E_{t+1}(s,a) = (1-\alpha_t(s,a))E_t(s,a) + \alpha_t(s,a)\left((\tilde{H}E_t)(s,a) + w_t(s,a)\right). \tag{3.4}$$

Dado que

$$\mathcal{W} = \{\omega \in \Omega : \lim_{t\to\infty} \|Q_t(\omega) - Q^*\|_{\mathcal{Q}} = 0\} = \{\omega \in \Omega : \lim_{t\to\infty} \|E_t(\omega)\|_{\mathcal{Q}} = 0\}$$

nuestra tarea es equivalente ahora a mostrar que E_t va a 0 con probabilidad 1.

Comentario: El operador $\tilde{H} : \mathcal{Q} \to \mathcal{Q}$ viene definido por

$$\tilde{H}E = H(E + Q^*) - Q^*.$$

Utilizando que H es contractivo, es directo concluir que $\tilde{H}$ también lo es y que, en su caso, el único punto fijo es el vector $Q \equiv 0$.

La última reflexión antes de analizar con detalle bajo qué condiciones se produce dicha convergencia es que cabe esperar que no todos los comportamientos del ruido $w_t(s,a)$ son admisibles en una iteración del tipo de la de (3.4). En lo que refiere al proceso asociado al algoritmo de Q-learning, se tiene que esa perturbación satisface las siguientes condiciones, que enunciamos como proposición pues nos referiremos a ellas de manera recurrente.

[3.2.8] Proposición. *Los vectores w_t definidos anteriormente satisfacen las siguientes relaciones, donde A y B son dos constantes mayores o iguales que cero.*

- $\mathbb{E}\left[w_t(s,a)\middle|\mathcal{F}_t\right] = 0$;
- $Var\left[w_t(s,a)\middle|\mathcal{F}_t\right] = \mathbb{E}\left[w_t(s,a)^2\middle|\mathcal{F}_t\right] \leq A + B\|E_t\|_{\mathcal{Q}}^2$.

[3.2.8] *Demostración.* La primera igualdad ya se ha discutido y se comentó que era una consecuencia trivial de la definición. Para ver la segunda, basta notar que $|w_t(s,a)|$ es igual a

$$\left| r(s,a,s'_t(s,a)) + \lambda \max_{b\in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b) - \sum_{j\in S} p(j|s,a)\left[r(s,a,j) + \lambda \max_{b\in A_j} Q_t(j,b)\right]\right|$$

Utilizando la desigualdad triangular para cada término llegamos a que

$$|w_t(s,a)| \leq M + \lambda\|Q_t\|_{\mathcal{Q}} + M + \lambda\|Q_t\|_{\mathcal{Q}},$$

y para terminar recordamos que $\|Q_t\|_{\mathcal{Q}} \leq \|E_t\|_{\mathcal{Q}} + \|Q^*\|_{\mathcal{Q}}$.

□

3.2.1. Teoremas de convergencia

Procedemos ahora a demostrar que la iteración (3.4) converge a 0 con probabilidad 1. Seguimos la idea de [4] (que a su vez sigue las ideas de [39]) en dos pasos: en primer lugar, mostramos que las sucesiones $\{E_t(s,a)(\omega)\}_{t\in\mathbb{N}}$ están acotadas con probabilidad 1, y a continuación razonamos que en ese caso se tiene que dar que $\lim_{t\to\infty} E_t(s,a)(\omega) = 0$ en casi todo ω. Durante toda esta subsección nos apoyaremos en lo discutido en el Apéndice B.

Comenzamos, pues, abordando primero la cuestión de la acotación de $\{E_t(s,a)(\omega)\}_{t\in\mathbb{N}_0}$. Para evitar la necesidad de introducir nueva notación que no volveremos a utilizar, enunciamos el teorema para unos vectores aleatorios que toman valores en el espacio de funciones $\mathcal{Q}$; no obstante, es evidente que todo lo que se discutirá a partir de ahora es válido para aproximaciones estocásticas con esta misma estructura en espacios vectoriales de dimensión finita.

[3.2.9] Teorema. *Supongamos que en una aproximación estocástica en la que los vectores aleatorios toman valores en $S \times A_s$ sus componentes $E_t(s,a)$ vienen dadas por*

$$E_{t+1}(s,a) = (1-\alpha_t(i))E_t(s,a) + \alpha_t(s,a)\Big[(\tilde{H}E_t)(s,a) + w_t(s,a)\Big].$$

donde se cumple que

1. *$\alpha_t(s,a) \in [0,1]$ y $\sum_{t=0}^{\infty}\alpha_t(s,a) = \infty$, $\sum_{t=0}^{\infty}\alpha_t^2(s,a) < \infty$ con probabilidad 1;*
2. *El ruido $\{w_t\}_{t\in\mathbb{N}_0}$ satisface las condiciones de la Proposición 3.2.8;*
3. *El operador $\tilde{H}$ es contractivo con constante $\lambda \in [0,1)$ y punto fijo 0.*

Entonces se verifica que las sucesiones $\{E_t(s,a)(\omega)\}_{t\in\mathbb{N}_0}$ están acotadas con probabilidad 1.

[3.2.9] *Demostración.* Comenzamos tomando un valor η tal que $\lambda < \eta < 1$, y llamaremos $\epsilon > 0$ al número tal que $\eta(1+\epsilon) = 1$; establecido esto, ya podemos pasar a introducir las variables fundamentales de la demostración.

La idea que vamos a seguir es básicamente introducir unos factores de escala $\{G_t\}_{t\in\mathbb{N}_0}$, donde las $G_t : \Omega \to \mathbb{R}_+$ son variables aleatorias. Éstas normalizaran el tamaño del error, de manera que podamos trabajar con unas variables aleatorias más sencillas que dan lugar a sucesiones $\{G_t(\omega)\}_{t\in\mathbb{N}_0}$ monótonas no decrecientes. Utilizando lo discutido en el Apéndice B podremos utilizar la convergencia de una variable auxiliar para concluir que la sucesión $\{G_t(\omega)\}$ está acotada.

Para definir esos factores de escala utilizamos la siguiente recursión:

- $G_0 = \max\{\|E_0\|_{\mathcal{Q}}, 1\}$;

- $G_{t+1} = G_t$ si $\|E_{t+1}\|_{\mathcal{Q}} \leq (1+\epsilon)G_t$,
- $G_{t+1} = G_0(1+\epsilon)^k$ si $\|E_{t+1}\|_{\mathcal{Q}} > (1+\epsilon)G_t$, siendo k el número natural tal que $G_0(1+\epsilon)^{k-1} < \|E_{t+1}\|_{\mathcal{Q}} \leq G_0(1+\epsilon)^k$.

De la definición de las v.a. $\{G_t\}_{t\in\mathbb{N}_0}$ deducimos que las G_t son $\mathcal{F}_t$-medibles, que efectivamente son monótonas no decrecientes y que $G_t \geq 1$. Además, se tiene también que

(3.5) 1. $\|E_t\|_{\mathcal{Q}} \leq (1+\epsilon)G_t$

(3.6) 2. $G_{t-1} < G_t \implies \|E_t\| \leq G_t$.

Usando esas igualdades, veamos ahora que $\|(\tilde{H}E_t)\|_{\mathcal{Q}} \leq G_t$; en efecto,

$$\begin{aligned}\|(\tilde{H}E_t)\|_{\mathcal{Q}} &\leq \lambda\|E_t\|_{\mathcal{Q}} \leq \lambda\|E_t\|_{\mathcal{Q}} + \eta - \lambda \leq \\ &\leq \lambda(1+\epsilon)G_t + (\eta-\lambda)G_t \leq G_t(\lambda(1+\epsilon) + (\eta-\lambda)) \leq \\ &\leq G_t(\lambda\epsilon + \eta) \leq G_t(\eta(1+\frac{\lambda}{\eta}\epsilon)) \leq G_t.\end{aligned}$$

Relacionemos ahora el tamaño del ruido $w_t(s,a)$ con nuestros factores de escala. Definimos

$$\tilde{w}_t(s,a) = \frac{w_t(s,a)}{G_t}$$

y vemos, recordando que G_t es medible, que

- $\mathbb{E}\left[\tilde{w}_t(s,a)|\mathcal{F}_t\right] = \frac{\mathbb{E}[w_t(s,a)|\mathcal{F}_t]}{G_t} = 0$;
- $\mathbb{E}\left[\tilde{w}_t^2(s,a)|\mathcal{F}_t\right] = \frac{\mathbb{E}\left[w_t^2(s,a)|\mathcal{F}_t\right]}{G_t^2} \leq \frac{A+B\|E_t\|_{\mathcal{Q}}^2}{G_t^2} \leq A + B(1+\epsilon)^2 = K$.

Definimos ahora los vectores aleatorios $\{\tilde{W}_{t;t_0}\}_{t\geq t_0}$ de modo que $\tilde{W}_{t_0;t_0}(s,a) = 0$ y

(3.7) $$\tilde{W}_{t+1;t_0}(s,a) = (1-\alpha_t(s,a))\tilde{W}_{t;t_0} + \alpha_t(s,a)\tilde{w}_t,\ t \geq t_0.$$

El Corolario B.0.2 nos va a permitir afirmar que estas variables aleatorias van a 0.

[3.2.10] Lema. *Con probabilidad 1, existe* $\forall\delta > 0$ *un* t_0 *tal que* $\left|\tilde{W}_{t;t_0}(s,a)\right| \leq \delta$ $\forall t \geq t_0$.

[3.2.10] *Demostración.* Por el Corolario B.0.2, se cumple que en el caso $t_0 = 0$ tenemos $\lim_{t\to\infty} \tilde{W}_{t;0}(s,a)(\omega) = 0$. Por otro lado, es fácil ver - usando la linealidad - que solo hay una posible solución a la definición (3.7). Dado que las variables

aleatorias

$$\left[\prod_{\tau=t_0}^{t-1}(1-\alpha_t(s,a))\right]\tilde{W}_{t;0}(s,a)+\tilde{W}_{t;t_0}(s,a),\ t\geq t_0$$

satisfacen también (3.7) y son iguales que $\tilde{W}_{t;0}(s,a)$ en t_0, necesariamente se cumple que para $t\geq t_0$

$$\tilde{W}_{t;0}(s,a)=\left[\prod_{\tau=t_0}^{t-1}(1-\alpha_t(s,a))\right]\tilde{W}_{t_0;0}(s,a)+\tilde{W}_{t;t_0}(s,a).$$

De ahí se concluye que

$$\begin{aligned}\left|\tilde{W}_{t;t_0}(s,a)\right|&\leq\left|\tilde{W}_{t;0}(s,a)\right|+\left|\left[\prod_{\tau=t_0}^{t-1}(1-\alpha_t(s,a))\right]\tilde{W}_{t_0;0}(s,a)\right|\leq\\&\leq\left|\tilde{W}_{t;0}(s,a)\right|+\left|\tilde{W}_{t_0;0}(s,a)\right|.\end{aligned}$$

Tomando t_0 de tal modo que $t\geq t_0\implies\left|\tilde{W}_{t;0}(s,a)(\omega)\right|\leq\frac{\delta}{2}$ concluimos.

□

Asumimos ahora que E_t no está acotado y llegamos a contradicción.

En efecto, si $\|E_t\|_{\mathcal{Q}}$ no está acotado se tiene por (3.5) que $G_t\to\infty$, y por (3.6) se tiene que $\|E_t\|_{\mathcal{Q}}\leq G_t$ para un cantidad infinita de valores de $t\in\mathbb{N}$. Este hecho, junto con el Lema 3.2.10, nos permite saber que para casi todo ω puedo encontrar un t_0 tal que $\|E_{t_0}\|_{\mathcal{Q}}\leq G_{t_0}$ y

$$|\tilde{W}_{t;t_0}(s,a)|\leq\epsilon\ \forall t\geq t_0\text{ y }(s,a)\in S\times A_S. \tag{3.8}$$

El siguiente lema da lugar a una contradicción que nos permite concluir.

[3.2.11] Lema. *Supongamos que existe un t_0 tal que $\|E_t\|_{\mathcal{Q}}\leq G_{t_0}$ y se cumple (3.8). Entonces, para $t\geq t_0$ se cumple que $G_t=G_{t_0}$ y para todo par $(s,a)\in S\times A_S$*

$$-G_{t_0}(1+\epsilon)\leq-G_{t_0}+\tilde{W}_{t;t_0}(s,a)G_{t_0}\leq E_t(s,a)\leq G_{t_0}+\tilde{W}_{t;t_0}(s,a)G_{t_0}\leq G_{t_0}(1+\epsilon).$$

[3.2.11] *Demostración.* Procedemos por inducción. Para $t=t_0$ es cierto por hipótesis y porque $\tilde{W}_{t_0;t_0}=0$. Por otro lado, si asumimos que es cierto para t, se

tiene que $G_t = G_{t_0}$ y vemos que

$$\begin{aligned} E_{t+1}(s,a) &= (1-\alpha_t(s,a))E_t(s,a) + \alpha_t(s,a)((\tilde{H}E_t)(s,a) + w_t(s,a)) \leq \\ &\leq (1-\alpha_t(s,a))(G_{t_0} + \tilde{W}_{t;t_0}G_{t_0}) + \alpha_t(s,a)((\tilde{H}E_t)(s,a) + G_{t_0}\tilde{w}_t(s,a)) \leq \\ &\leq (1-\alpha_t(s,a))(G_{t_0} + \tilde{W}_{t;t_0}G_{t_0}) + \alpha_t(s,a)(G_{t_0} + G_{t_0}\tilde{w}_t(s,a)) \leq \\ &\leq G_{t_0} + G_{t_0}((1-\alpha_t(s,a))\tilde{W}_{t;t_0} + \alpha_t(s,a)\tilde{w}_t) = \\ &= G_{t_0} + G_{t_0}\tilde{W}_{t+1;t_0}. \end{aligned}$$

Las otras desigualdades se obtienen de manera totalmente análoga.

□

□

A partir de aquí es posible demostrar el siguiente teorema, de donde ya sí se podrá deducir la convergencia del método.

[3.2.12] Teorema. *Sea E_t la sucesión de vectores aleatorios con condición inicial determinista $E_0 \in \mathcal{Q}$ y cuyas componentes vienen definidas del siguiente modo*

$$E_{t+1}(s,a) = (1-\alpha_t(s,a))E_t(s,a) + \alpha_t(s,a)\Big[(\tilde{H}E_t)(s,a) + w_t(s,a)\Big].$$

Suponemos que se satisface que

1. *$\alpha_t(s,a) \in [0,1]$ y $\sum_{t=0}^{\infty}\alpha_t(s,a) = \infty$, $\sum_{t=0}^{\infty}\alpha_t^2(s,a) < \infty$ con probabilidad 1;*
2. *El ruido $\{w_t\}_{t\in\mathbb{N}}$ satisface las condiciones de la Proposición 3.2.8;*
3. *El operador $\tilde{H}$ es contractivo con constante $\lambda \in [0,1)$ y punto fijo 0.*

Entonces E_t converge a 0 con probabilidad 1.

[3.2.12] *Demostración.* Lo primero que observamos es que se satisfacen todas las condiciones del Teorema 3.2.9. Por tanto, existe una variable aleatoria D_0 tal que $D_0 < \infty$ con probabilidad 1 tal que $\|E_t\|_{\mathcal{Q}} \leq D_0$ para todo t. Establecido esto, fijamos un $\epsilon > 0$ tal que $\lambda + \epsilon < 1$ y vamos a definir la sucesión

$$D_{k+1} = (\lambda + \epsilon)D_k,\ k \geq 0.$$

Es evidente que $\lim_{k\to\infty} D_k = 0$ con probabilidad 1.

Queremos mostrar que en casi todo ω vamos a poder encontrar, para todo $k \in \mathbb{N}_0$, un t_k tal que

$$\|E_t\|_{\mathcal{Q}} \leq D_k\ \forall t \geq t_k. \tag{3.9}$$

En efecto, esto nos permitiría concluir que la afirmación que perseguimos demostrar es cierta, pues sabríamos que

$$\limsup_{t\to\infty} \|E_t\|_{\mathcal{Q}} \leq D_k \text{ con probabilidad 1 para todo } k \in \mathbb{N}_0$$

Dado que $\inf_{k\in\mathbb{N}_0} D_k = 0$ en casi todo ω, habríamos concluido que se da la convergencia que queríamos con probabilidad 1.

Para probar la afirmación anterior, procedemos por inducción, notando que el caso $k = 0$ es cierto con $t_0 = 0$. El objetivo de la demostración a partir de ahora se va a reducir simplemente a ver que la asunción de que (3.9) se cumple para un k nos permite concluir que también es cierta para $k + 1$.

Con este objetivo, comenzamos definiendo, de manera similar al teorema anterior, las familias de vectores aleatorios $\{W_{t;\tau}\}_{t\geq\tau}$ de modo que $W_{\tau;\tau}(s, a) = 0$ y

$$W_{t+1;\tau}(s, a) = (1 - \alpha_t(s, a))W_{t;\tau} + \alpha_t(s, a)w_t(s, a).$$

Utilizando el hecho de que $\|E_t\|_{\mathcal{Q}} \leq D_0$, observamos que

$$\mathbb{E}\left[w_t(s, a)^2 \middle| \mathcal{F}_t\right] \leq A + B\|E_t\|^2_{\mathcal{Q}} \leq A + BD_0^2.$$

De este modo, se cumplen las hipótesis del Corolario B.0.2, por lo que deducimos que

$$\lim_{t\to\infty} W_{t;0}(s, a) = 0 \text{ en casi todo } \omega.$$

Del mismo modo, razonando como en el Lema 3.2.10 y recordando (ver [38]) que

$$\prod_{k=\tau}^{\infty} [1 - \alpha_k(s, a)] = 0 \iff \sum_{k=\tau}^{\infty} \alpha_k(s, a) = \infty,$$

concluimos también que para todo τ

$$\lim_{t\to\infty} W_{t;\tau}(s, a) = 0 \text{ en casi todo } \omega.$$

Introducimos ahora también la familia de vectores aleatorios $\{Y_{t;t_k}\}_{t\geq t_k}$, definidos de acuerdo a $Y_{t_k;t_k}(s, a) = D_k$ y

$$Y_{t+1:t_k}(s, a) = (1 - \alpha_t(s, a))Y_{t;t_k} + \alpha_t(s, a)\lambda D_k.$$

Para ver el comportamiento de $Y_{t;t_k}(s, a)$ cuando t va a infinito, restamos en ambos lados de la igualdad anterior el término λD_k y vemos entonces que $\hat{Y}_{t;t_k}(s, a) = Y_{t;t_k}(s, a) - \lambda D_k$ cumple que

$$\hat{Y}_{t+1;t_k}(s, a) = (1-\alpha_t(s, a))\hat{Y}_{t;t_k}(s, a) \implies \hat{Y}_{t;t_k}(s, a) = \left[\prod_{j=t_k}^{t-1}(1 - \alpha_j(s, a))\right](1-\lambda)D_k.$$

Como $\sum_{j=0}^{\infty} \alpha_j(s,a) = \infty$ se deduce (ver, de nuevo, [38]) que $\lim_{t\to\infty} \hat{Y}_{t;t_k}(s,a) = 0$ en los ω con $D_0 < \infty$ y, por tanto, $\lim_{t\to\infty} Y_{t;t_k}(s,a) = \lambda D_k$ con probabilidad 1.

La utilidad de estos vectores aleatorios queda patente en el siguiente lema.

[3.2.13] Lema. *En todo par $(s,a) \in S \times A_S$ se cumple que para $t \geq t_k$*

$$-Y_{t;t_k}(s,a) + W_{t;t_k}(s,a) \leq E_t(s,a) \leq Y_{t;t_k}(s,a) + W_{t;t_k}(s,a).$$

[3.2.13] *Demostración.* Una vez más, basta aplicar inducción. El resultado es obviamente cierto si $t = t_k$. Si suponemos ahora que es cierto para algún $t \geq t_k$, vemos que

$$\begin{aligned} E_{t+1}(s,a) &\leq (1-\alpha_t(s,a))(Y_{t;t_k}(s,a) + W_{t;t_k}(s,a)) + \alpha_t(s,a)((\tilde{H}E_t)(s,a) + w_t(s,a)) \leq \\ &\leq (1-\alpha_t(s,a))(Y_{t;t_k}(s,a) + W_{t;t_k}(s,a)) + \alpha_t(s,a)(\lambda D_k + w_t(s,a)) = \\ &= Y_{t+1;t_k}(s,a) + W_{t+1;t_k}(s,a). \end{aligned}$$

La otra desigualdad se obtiene de manera totalmente análoga.

□

Puesto que en casi todo ω se tiene que $\lim_{t\to\infty} Y_{t;t_k}(s,a) = \lambda D_k$ y $\lim_{t\to\infty} W_{t;t_k}(s,a) = 0$, deducimos que con probabilidad 1 y para todo par $(s,a) \in S \times A_S$

$$\limsup_{t\to\infty} |E_t(s,a)| \leq \lambda D_k < (\lambda + \epsilon) D_k = D_{k+1}.$$

De este modo, como $|S \times A_S| < \infty$,

$$\limsup_{t\to\infty} \|E_t\|_{\mathcal{Q}} \leq \lambda D_k < (\lambda + \epsilon) D_k = D_{k+1}.$$

Por tanto, existe un tiempo t_{k+1} tal que $\|E_t\| \leq D_{k+1}$ para todo $t \geq t_{k+1}$; esto es justo lo que nos permite concluir que la inducción que queríamos demostrar es correcta.

□

3.2.2. Convergencia de Q-learning

Utilizando estos resultados, se puede concluir de manera directa que - si se toma el valor de T suficientemente grande - el siguiente algoritmo da lugar a una función $a_q \in A_S^S$ que representa una política estacionaria óptima con probabilidad 1.

Algoritmo de Q-learning

Consideramos una familia de vectores aleatorios $\{\alpha_t\}_{t\in\mathbb{N}_0}$ $\mathcal{F}_{t+1}$-medibles y una condición inicial arbitraria $Q_0 \in \mathcal{Q}$. Suponemos que $|r(s,a,j)| \leq M$ y que se cumple para todo (s,a) con probabilidad 1 que $\alpha_t(s,a) \in [0,1]$ y

$$\sum_{t=0}^{\infty} \alpha_t(s,a) = \infty, \quad \sum_{t=0}^{\infty} \alpha_t(s,a)^2 < \infty.$$

1. Definimos iterativamente los vectores $\{Q_t\}_{0\leq t\leq T}$

$$Q_{t+1}(s,a) = (1-\alpha_t(s,a))Q_t(s,a)+$$

$$+\alpha_t(s,a)\Big[r(s,a,s'_t(s,a)) + \lambda \max_{b\in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b)\Big].$$

2. Escogemos al función $a_{Q-l} : S \to A$ de acuerdo a

$$a_{Q-l}(s) = \arg\max_{a\in A_S} Q_T(s,a).$$

[3.2.14] Teorema. (Convergencia de Q-learning). *Si T es suficientemente grande, con el algoritmo anterior obtenemos con probabilidad 1 una función $a_{Q-l} \equiv a_q^*$, siendo a_q^* un mapa asociado a una política estacionaria determinista óptima.*

[3.2.14] *Demostración.* Tal y como ya argumentamos, la sucesión $\{Q_t\}_{t\in\mathbb{N}_0}$ converge con probabilidad 1 a Q^* si y solo si

$$E_{t+1} = (1-\alpha_t(i))E_t(s,a) + \alpha_t(s,a)\Big((\tilde{H}r_t)(s,a) + w_t(s,a)\Big)$$

converge a 0 con probabilidad 1. En el caso de Q-learning presentado aquí, el operador $\tilde{H}$ es contractivo y el ruido $\{w_t\}_{t\in\mathbb{N}_0}$ satisface las conclusiones de la Proposición 3.2.8. Por la hipótesis sobre los vectores aleatorios $\{\alpha_t\}_{t\in\mathbb{N}_0}$, observamos que se cumplen las condiciones de los Teoremas 3.2.9 y 3.2.12, por lo que los vectores aleatorios E_t tienden en la norma $\mathcal{Q}$ al vector 0 con probabilidad 1 y, por ende, los vectores Q_t tienden a Q^* en norma también con probabilidad 1.

Definimos ahora los valores ϵ_s para cada $s \in S$ del siguiente modo

$$\epsilon_s := \min\{|Q^*(s,a_1) - Q^*(s,a_2)| : a_1 \in A_s, a_2 \in A_s, |Q^*(s,a_1) - Q^*(s,a_2)| > 0\}.$$

Fijamos ahora $\epsilon > 0$ tal que $\epsilon < \min_{s\in S} \frac{\epsilon_s}{2}$. Dado que Q_t tiende a Q^* en norma, es posible encontrar para casi todo ω un valor $T^*(\omega)$ tal que $t \geq T^* \implies \|Q_t - Q^*\|_{\mathcal{Q}} < \epsilon$. Así pues, si $T \geq T^*$ es sencillo ver que por construcción la función a_{Q-l} que tome el algoritmo necesariamente ha de coincidir con una función a_q^* tal que $Q^*(s, a_q^*(s)) =$

$\max_{a \in A_s} Q^*(s,a)$. Por el Teorema 3.1.2, concluimos que la función $a_{Q-l} \equiv a_q^*$ da lugar a una política estacionaria determinista óptima.

□

Comentario: A la hora de llevar a cabo una estimación de la política óptima usando Q-learning, observamos que el usuario debe seleccionar la condición inicial Q_0 y la familia de vectores aleatorios $\{\alpha_t\}_{t \in \mathbb{N}_0}$. En los próximos párrafos, discutiremos cómo estos valores vendrán en ocasiones parcialmente determinados por la aplicación en concreto y el método de exploración que se esté utilizando.

3.2.3. Paradigmas básicos de exploración

La primera conclusión obvia del Teorema 3.2.14 es que, tal y como se deduce de la condición $\sum_{t=0}^{\infty} \alpha_t(s,a) = \infty$, para tener garantizada la convergencia requerimos que todos los pares estado-acción $(s,a) \in S \times A_S$ sean visitados una cantidad infinita de veces con probabilidad 1. Ésta es una condición intuitivamente insoslayable; debido al carácter probabilístico del modelo subyacente, es fácil ver que no es posible en general obtener el valor Q^* con probabilidad 1 en un número finito de iteraciones.

No obstante, el teorema nos deja abierta la cuestión de cómo *organizar* esas visitas mediante la libertad que nos otorga a la hora de seleccionar la forma de los vectores $\{\alpha_t\}_{t \in \mathbb{N}_0}$. En efecto, es a través de éstos valores como se determina qué pares estado-acción se han explorado en el instante t del algoritmo: si en t se ha observado el resultado del par (s,a), se conocerá el valor de $s'_t(s,a)$ y es posible tomar $\alpha_t(s,a)(\omega) \neq 0$. Si por otro lado en t no se ha podido conocer el resultado de ejercer la acción a en el estado s, la manera de introducir esto en nuestro proceso estocástico es forzar que $\alpha_t(s,a)(\omega) = 0$. En base a la cantidad de componentes no nulas de α_t se establece la primera clasificación de algoritmos Q-learning: si con probabilidad estrictamente positiva se tiene que para algún instante $t \in \mathbb{N}_0$ se verifica que más de un valor $\alpha_t(s,a) \neq 0$, diremos que es un algoritmo sincrónico, mientras que si con probabilidad 1 todos los $\alpha_t(s,a) = 0$ excepto uno se dirá que el algoritmo es asíncrono.

Los algoritmos sincrónicos suelen ser más propios de situaciones en las que disponemos de observaciones acumuladas, como cuando se usan bases de datos. En este tipo de casos, es natural suponer que conocemos de antemano cuáles son los pares estado-acción, por lo que es posible tomar en todo instante t una de las observaciones acumuladas para cada par (s,a) y asociarla a la variable aleatoria $s'_t(s,a)$; de este modo, tiene sentido considerar que se corre el algoritmo tomando el mismo $\alpha_t(s,a) = \alpha(t)$ (con $\sum_{t=0}^{\infty} \alpha(t) = \infty$, $\sum_{t=0}^{\infty} \alpha(t)^2 < \infty$) para todos los elementos de $S \times A_S$ a la vez. Evidentemente, en este caso en el que todos los pares se visitan para todo t se satisfacen las condiciones del Teorema 3.2.14, por lo que tenemos garantizado que esta manera de proceder tiene límite; a este método se le conoce en la literatura como *Pa-*

rallel Sampling y se utiliza en ocasiones para modelar el caso de exploración perfecta y centrarse así exclusivamente en cuestiones relacionadas con la convergencia. Desde el punto de vista computacional, este paradigma presenta el atractivo de ser fácilmente paralelizable.

Por otro lado, los algoritmos asíncronos son característicos en situaciones en las que el agente no dispone de ninguna información previa sobre el sistema (quizá ni siquiera una visión global de todos los pares (s,a)) y solo tiene la posibilidad de realizar simulaciones. En ese caso, si se desea ir incorporando la información según se recibe, notamos que en el instante t solo hemos podido observar el valor de la variable aleatoria $s'_t(s,a)$ para el par en el que ha tenido lugar la simulación; es natural, pues, utilizar el algoritmo de Q-learning con $\alpha_t(s,a) \neq 0$ en el par (s,a) donde hemos simulado y $\alpha_t(s,a) = 0$ en los demás.

Para garantizar que esa metodología converja, vuelve a ser necesario que todos los pares sean visitados una cantidad infinita de veces. Es en este instante donde se puede plantear adecuadamente el problema de la exploración: nuestro objetivo es visitar de la manera más eficiente un sistema que no conocemos. Ésta es una cuestión que ha concentrado una cantidad significativa de esfuerzos y que ha dado lugar a métodos matemáticamente sofisticados, aunque - como se indica en [36] - "en muchas ocasiones descansan en asunciones imposibles de verificar en la práctica". A modo de ejemplo, una asunción habitual que se suele hacer sobre el proceso de decisión de Markov subyacente es que sea una *unichain*, lo cuál exige que para toda política $\pi \in \Pi^{MD}$ la cadena de Markov resultante está formada por una única clase recurrente y un conjunto de estados transitorios. Como se puede ver, de cara a las aplicaciones no es consistente asumir que no conocemos siquiera cuáles son los pares estado-acción del modelo pero a su vez exigir que el MDP subyacente satisfaga una condición tal.

En esta subsección nos limitaremos, pues, a exponer brevemente los dos métodos exploratorios más habituales. Para asegurar que se puedan cumplir las hipótesis del Teorema 3.2.14, comenzamos asumiendo que cada L iteraciones el agente se reubicará de manera aleatoria uniformemente distribuida en cualquiera de los estados $s \in S$; de este modo, no es difícil demostrar que esta condición es suficiente para al menos garantizar que con probabilidad 1 se van a visitar todos los estados una cantidad infinita de veces.

La primera técnica de exploración que presentamos es la que se conoce como *avariciosa* (en inglés, *greedy*). Se basa en intentar explotar, en base a la aproximación de la que se dispone en el instante t, la acción que se considera más beneficiosa.

Exploración *avariciosa*

Denotamos por $s_g(t)$ el estado en el que se encuentra el agente en el instante t, y fijamos $Q_0 \in \mathcal{Q}$. Sea $\{\alpha(t)\}_{t\in\mathbb{N}_0} \subset [0,1]$ una sucesión tal que $\sum_{t=0}^{\infty} \alpha(t) = \infty$ y $\sum_{t=0}^{\infty} \alpha(t)^2 < \infty$ con probabilidad 1.

1. Tomamos para $s_g(0)$ un valor aleatorio uniformemente distribuido en S;
2. Para $t \geq 0$,
   a) Escogemos $a_g(t)$ de modo que $a_g(t) \in \arg\max_{a\in A_{s_g(t)}} Q_t(s_g(t), a)$;
   b) Realizamos una simulación en el par $(s_g(t), a_g(t))$ y observamos el valor $s'_t(s_g(t), a_g(t))$;
   c) Tomamos $\alpha_t(s,a) = 0$ si $(s,a) \neq (s_g(t), a_g(t))$ y $\alpha_t(s_g(t), a_g(t)) = \alpha(\#[s_g(t), a_g(t), t])$, donde $\#[s_g(t), a_g(t), t]$ representa el número de veces que se ha visitado el par $(s_g(t), a_g(t))$ en lo que va de algoritmo antes del instante t. Claramente, las $\alpha_t(s,a)$ son $\mathcal{F}_t$-medibles;
   d) Realizamos la iteración de Q-learning;
   e) Seleccionamos $s_g(t+1)$:
      1) Si $(t+1)$ mód $L = 0$, escogemos un valor aleatorio uniformemente distribuido en S para $s_g(t+1)$;
      2) Si $(t+1)$ mód $L \neq 0$, tomamos $s_g(t+1) = s'_t(s_g(t), a_g(t))$.

La idea que fundamenta la exploración *avariciosa* es explotar y extraerle todo el jugo posible al conocimiento que ya se tiene. Por la manera en la que se selecciona $s_g(t+1)$, observamos que esta técnica va a observar y actualizar con mayor frecuencia las secuencias de estados que ya considera las mejores. Por tanto, se supone que de este modo se centrará más la atención en los estados más *importantes*, aquellos donde me suelen llevar políticas aproximadamente óptimas. Utilizando el ejemplo del agente que busca salir de una habitación, es de esperar que este tipo de exploraciones visite con mucha asiduidad los estados cercanos a la puerta, mientras que las esquinas alejadas solamente se observarán cuando se reubique al agente de manera aleatoria.

El problema de este enfoque es que, dado un estado, el algoritmo podría *viciarse* con una acción que en su aproximación considera la mejor para esa situación, y en principio no habría motivos para que explore otras posibilidades que quizá le acaben llevando a mejor puerto. Siguiendo con el ejemplo anterior, podría ocurrir que el agente ha encontrado una manera de salir de la habitación y se contenta con ello, por lo que deja de considerar la posibilidad de buscar otras puertas más cercanas. Así, por mucho que la aleatoriedad que imponemos cada L pasos nos permita visitar todos los estados una cantidad infinita de veces, podría ocurrir que haya pares (s,a) que solo se ejecuten en una cantidad finita de iteraciones.

En cierto modo, el fenómeno subyacente que está ocurriendo aquí es similar a lo que se observa con ciertos métodos de optimización basados en el gradiente cuando se quedan *atascados* en un mínimo local. A pesar de la aleatoriedad que se introduce en el espacio de estados cada L pasos, se puede llegar a una configuración de *equilibrio* para las acciones que se escogen en cada estado s, de tal modo que realizar pequeñas exploraciones de nuevos pares (s, a) no parezcan inducir mejorías. Como es sabido, la manera de abordar estas situaciones es complementar esas técnicas de búsqueda con otra componente aleatoria que me obligue a visitar con cierta asiduidad otras acciones: esto da lugar a lo que se conoce como exploración ϵ-*avariciosa*.

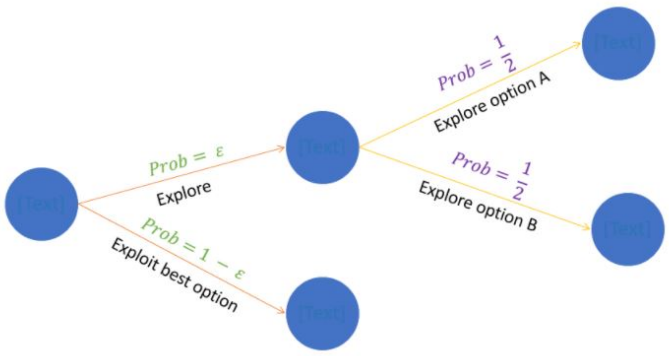


Figura 3.2: Esquema de la elección de $a_g(t)$ de acuerdo a una exploración ϵ-avariciosa

Exploración ϵ-*avariciosa*

Denotamos por $s_g(t)$ el estado en el que se encuentra el agente en el instante t, y fijamos $Q_0 \in \mathcal{Q}$. Sea $\{\alpha(t)\}_{t\in\mathbb{N}_0} \subset [0,1]$ una sucesión tal que $\sum_{t=0}^{\infty} \alpha(t) = \infty$ y $\sum_{t=0}^{\infty} \alpha(t)^2 < \infty$ con probabilidad 1. Escogemos un $\epsilon \in (0,1)$ que controla la aleatoriedad a la hora de escoger acciones

1. Tomamos para $s_g(0)$ un valor aleatorio uniformemente distribuido en S;
2. Para $t \geq 0$,
   - *a*) Con probabilidad $1-\epsilon$, escogemos $a_g(t)$ de modo que $a_g(t) \in \arg\max_{a\in A_{s_g(t)}} Q_t(s_g(t), a)$, y con probabilidad ϵ tomamos aleatoriamente de manera uniforme $a_g(t) \in A_{s_g(t)} \setminus \arg\max_{a\in A_{s_g(t)}} Q_t(s_g(t), a)$;
   - *b*) Realizamos una simulación en el par $(s_g(t), a_g(t))$ y observamos el valor $s'_t(s, a)$;
   - *c*) Tomamos $\alpha_t(s, a) = 0$ si $(s, a) \neq (s_g(t), a_g(t))$ y $\alpha_t(s_g(t), a_g(t)) = \alpha(\#[s_g(t), a_g(t), t])$, donde $\#[s_g(t), a_g(t), t]$ representa el número de veces que se ha visitado el par $(s_g(t), a_g(t))$ en lo que va de algoritmo antes del instante t. De nuevo, las $\alpha_t(s, a)$ son $\mathcal{F}_t$-medibles;

d) Realizamos la iteración de Q-learning;

e) Seleccionamos $s_g(t+1)$:

1) Si $(t+1)$ mód $L = 0$, escogemos un valor aleatorio uniformemente distribuido en S para $s_g(t+1)$;
2) Si $(t+1)$ mód $L \neq 0$, tomamos $s_g(t+1) = s'_t(s,a)$.

De manera similar a como se argumentaba antes, se puede demostrar que la componente de aleatoriedad al escoger las acciones $a_g(t)$ va a conseguir que se visiten todos los pares $(s,a) \in S \times A_S$ con probabilidad 1, de tal modo que ahora sí tendríamos garantizada - al menos de manera teórica, pues en la práctica puede exigir una cantidad inasequible de pasos - la convergencia en todo par. Así, este enfoque nos permite incorporar un híbrido entre una técnica puramente exploratoria (que nos garantiza una exploración que posibilita la convergencia) y un método *avaricioso* ($\epsilon = 0$, que en principio nos ayudaría a visitar con más frecuencia y caracterizar mejor aquellas situaciones que más recompensa otorgan). Para clausurar la analogía con los métodos de optimización, el intento de equilibrar explotación vs. exploración equivale en optimización al balance entre esfuerzos de búsqueda local y búsqueda global.

3.3. Estimaciones en muestras finitas

Recapitulando lo establecido hasta ahora, hemos concluido que el algoritmo de Q-learning (ya sea sincrónico o asíncrono) converge adecuadamente a la función Q^* cuando (1) el número de iteraciones tiende a infinito, (2) se actualiza una cantidad infinita de veces cada par $(s,a) \in S \times A_S$ y (3) los coeficientes de aprendizaje $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ satisfacen con probabilidad 1 que

- $\sum_{t=0}^{\infty} \alpha_t(s,a) = \infty$;
- $\sum_{t=0}^{\infty} \alpha_t(s,a)^2 < \infty$.

Este tipo de resultados, si bien cruciales desde un punto de vista fundamental (pues nos garantizan que en última instancia el método funciona), requieren de complementos de cara a caracterizar y abordar adecuadamente las aplicaciones numéricas. En efecto, nótese que nuestros teoremas solo hacen mención a las colas de las sucesiones $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$, mientras que en un ensayo computacional solo se podrá *observar* una sucesión truncada $\{\alpha_t(s,a)\}_{0\leq t\leq T}$. Sobre la elección de esa cantidad grande pero finita de valores en principio no se ha impuesto nada, pero es claro que su influencia será capital en los resultados que se obtendrán al correr el algoritmo en números finitos de pasos.

Por otro lado, de cara a las aplicaciones es fundamental también disponer de cotas que nos indiquen la manera en la que crece el esfuerzo computacional según deseemos aumentar la precisión o variemos los valores de $|S|$, $|A|$ y λ. Este tipo de cotas serán, como consecuencia de lo discutido en el párrafo anterior, dependientes de los valores de $\{\alpha_t(s,a)\}_{0\leq t\leq T}$. En algunos casos, como veremos a continuación y siguiendo una metodología habitual en cálculo numérico, será útil parametrizar esa familia de valores $\{\alpha_t(s,a)\}_{0\leq t\leq T}$ de acuerdo a una variable β, encontrar algún tipo de cota para ese caso particular del algoritmo Q-learning y estudiar el comportamiento de los distintos términos involucrados en la estimación según se consideran múltiples valores de β.

Respecto a la naturaleza de las estimaciones que se van a obtener, cabe destacar que el carácter aleatorio del algoritmo hace que no sea posible garantizar que, para un número T de pasos, el tamaño del error esté contenido en un cierto intervalo; no se puede descartar la posibilidad de que en esa realización concreta puedan sucederse valores de $\{s'_t(s,a)\}_{0\leq t\leq T}$ fuera de lo *común*. En su lugar, será necesario hablar, como es habitual en entornos probabilísticos, de número de iteraciones hasta que $\|E_t\|_{\mathcal{Q}} \leq \epsilon$ con probabilidad $1-\delta$.

Estudios sobre el comportamiento en tiempo finito del algoritmo Q-learning se han llevado a cabo en trabajos como [19] o [37]. A modo de ejemplo, supongamos que se considera el algortimo de Q-learning sincrónico con una de las primeras sucesiones $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ en las que uno puede pensar: $\alpha_t(s,a) = \frac{1}{t+1}$ para todo $(s,a) \in S \times A_S$. En ese caso, en [19] se demuestra el siguiente resultado.

[3.3.15] Teorema. *Fijamos dos números $\delta \in (0,1)$ y $\epsilon > 0$ y consideremos la aplicación del algoritmo de Q-learning con un aprendizaje sincrónico tal que $\alpha_t(s,a) = \frac{1}{t+1}$ en todo $(s,a) \in S \times A_S$. Entonces, con probabilidad al menos $1-\delta$, se tiene que el error satisface $\|Q_T - Q^*\|_{\mathcal{Q}} \leq \epsilon$ si el valor de T es tal que*

$$T > C\left(3^{\frac{2}{1-\lambda}\ln\left(\frac{V_{max}}{\epsilon}\right)} \frac{V_{\text{máx}}^2 \ln\left(\frac{|S||A|V_{max}}{\delta(1-\lambda)\epsilon}\right)}{((1-\lambda)\epsilon)^2}\right), \tag{3.10}$$

donde $C > 0$ es una constante y V_{max} es

$$V_{max} = \frac{\sup_{(s,a,j)\in S\times A\times S}|r(s,a,j)|}{1-\lambda}.$$

[3.3.15] *Bosquejo de la demostración.* La idea consiste en utilizar unas desigualdades similares a la deducidas en el Lema 3.2.13,

$$-Y_{t;t_k}(s,a) + W_{t;t_k}(s,a) \leq E_t(s,a) \leq Y_{t;t_k}(s,a) + W_{t;t_k}(s,a),$$

que se cumplan para $t \geq t_k$, y encontrar un valor de t_{k+1} que nos permita controlar los valores de $Y_{t;t_k}(s,a)$ y $W_{t;t_k}(s,a)$ para $t \geq t_{k+1}$ de modo que $t \geq t_{k+1} \implies \|E_t\|_{\mathcal{Q}} \leq$

D_{k+1}. El término $Y_{t;t_k}(s,a)$ se acota fácilmente en un número finito de pasos pues su evolución es determinista. Para estimar $W_{t;t_k}(s,a)$ en $t \geq t_{k+1}$, se expresa como una martingala y se emplea la desigualdad de Azuma.

□

Otro caso que se puede considerar teóricamente es el de las sucesiones $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ dadas por $\alpha_t(s,a) = \frac{1}{(t+1)^\beta}$, con $\beta \in (\frac{1}{2}, 1)$. En este caso, se obtendría la siguiente expresión.

[3.3.16] Teorema. *Fijamos dos números $\delta \in (0,1)$ y $\epsilon > 0$ y consideramos la aplicación del algoritmo de Q-learning con un aprendizaje sincrónico tal que $\alpha_t(s,a) = \frac{1}{(t+1)^\beta}$ en todo $(s,a) \in S \times A_S$, con $\beta \in (\frac{1}{2}, 1)$. Entonces, con probabilidad al menos $1-\delta$, se tiene que el error satisface $\|Q_T - Q^*\|_{\mathcal{Q}} \leq \epsilon$ si el valor de T es tal que*

$$T > C\left(\left(\frac{V_{\text{máx}}^2 \ln\left(\frac{|S||A|V_{max}}{\delta(1-\lambda)\epsilon}\right)}{((1-\lambda)\epsilon)^2}\right)^{\frac{1}{\beta}} + \left(\frac{1}{1-\lambda}\ln\frac{V_{max}}{\epsilon}\right)^{\frac{1}{1-\beta}}\right), \tag{3.11}$$

donde, de nuevo, $C > 0$ es una constante y V_{max} es

$$V_{max} = \frac{\sup_{(s,a,j)\in S\times A\times S}|r(s,a,j)|}{1-\lambda}.$$

[3.3.16] *Bosquejo de la demostración.* La demostración se realiza de manera similar al teorema anterior, que recordamos que consistía en utilizar una desigualdad del tipo

$$-Y_{t;t_k}(s,a) + W_{t;t_k}(s,a) \leq E_t(s,a) \leq Y_{t;t_k}(s,a) + W_{t;t_k}(s,a),$$

y ver que $|Y_{t;t_k}(s,a) + W_{t;t_k}(s,a)| \leq |Y_{t;t_k}(s,a)| + |W_{t;t_k}(s,a)| \leq D_{k+1}$ si $t \geq t_{k+1}$. El primer término de la cota viene de encontrar el t_0 más pequeño para el cuál los $|W_{t;t_k}(s,a)|$ son lo suficientemente estables para poder satisfacer esta condición para todos los $t_0, t_1, ..., t_k$; a partir de ahí, habría que realizar el proceso para $t_0, t_1, ...$ hasta t_m, donde ya se cumpliría que $D_m \leq \epsilon$, y tomar $T \geq t_m$. El segundo sumando está directamente relacionado con ese número m; *grosso modo* refleja el valor $t_m - t_0$.

□

Las cotas anteriores incitan a pensar que el coste computacional asociado al algoritmo de Q-learning sincrónico utilizando esos valores de $\{\alpha_t(s,a)\}_{t\in N_0}$ es excesivo: por ejemplo, para el caso $\alpha_t(s,a) = \frac{1}{t+1}$, si fijamos $V_{\text{máx}} = 1$, $\epsilon = 0{,}01$ (que, para ese valor de $V_{\text{máx}}$, supondría exigir un error relativo de alrededor del 1 %) y $\lambda = 0{,}9$, se obtiene que solamente el primer factor de la estimación requeriría que $T \approx 3^{20}$. Inspeccionando la demostración y observando con detalle de donde surge cada término,

se observan al menos dos motivos que dan lugar a una cota teórica tan grande.

Para desarrollar esas dos causas, comenzamos comentando que el primer factor de la expresión (3.10) resulta de acotar los procesos deterministas $\{Y_{t;t_k}\}_{t\geq t_k}$, mientras que el segundo resulta de controlar de manera probabilística los procesos $\{W_{t;t_k}\}_{t\geq t_k}$. Establecido esto, el primer motivo por el que la cota es grande es una cuestión fundamental: que el término determinista solo se pueda controlar con unos valores de T tan grandes nos insinúa que los factores de aprendizaje $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ decrecen demasiado rápido e introducen un exceso de estabilidad que dificulta la convergencia. Así pues, una manera de solucionar este asunto sería tomar, por ejemplo,

$$\alpha_t(s,a) = \frac{1}{1+\left\lfloor \frac{t}{K} \right\rfloor} \tag{3.12}$$

con $K \in \mathbb{N}$, de modo que se favorece la convergencia de los procesos deterministas $\{Y_{t;t_k}\}_{t\geq t_k}$ (disminuyendo el primer factor) a cambio de empeorar el control de los procesos $\{W_{t;t_k}\}_{t\geq t_k}$ (es decir, aumentar el segundo factor). Observando los tamaños relativos de los dos factores vemos que sí es recomendable realizar este intercambio al menos hasta que se modifique en exceso el carácter de ambos términos. Este razonamiento, sugerido por el estudio de la demostración de la cota, se encuentra respaldado por las observaciones empíricas en ensayos numéricos: si se toma $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0} = \{\frac{1}{1+t}\}_{t\in\mathbb{N}_0}$ la evolución del error $\|E_t\|_{\mathcal{Q}}$ es excesivamente suave, mientras que considerando valores del tipo $\frac{1}{1+\left\lfloor \frac{t}{K} \right\rfloor}$ se contempla que aparecen oscilaciones debidas al ruido pero que en agregado la convergencia es mejor (ver Figura 3.3)

Por otro lado, el segundo motivo por el que la cota del Teorema 3.3.15 es notablemente grande se debe a una cuestión puramente técnica en la derivación de (3.10): llegado un cierto punto, se puede ver que la cota ha de ser el máximo entre - *grosso modo* - el segundo factor y algo proporcional al primer factor. Con el objeto de obtener una expresión simple y a la vista de que no es fácil ver cuál es más grande en general, en el artículo [19] se toma el producto de ambos. No obstante, si se toman unos valores de $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ como los de la expresión (3.12), se puede observar que el primer factor no se va a disparar tanto mientras se intuye que el segundo se va a mantener un comportamiento similar; así pues, considerando el máximo entre ambos se llegará a que

$$T = \mathcal{O}\left(\frac{V_{\text{máx}}^2 \ln\left(\frac{|S||A|V_{max}}{\delta(1-\lambda)\epsilon} \right)}{((1-\lambda)\epsilon)^2} \right).$$

De este modo, escogiendo valores de $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ del tipo

$$\left\{ \frac{1}{1+\left\lfloor \frac{t}{K} \right\rfloor} \right\}_{t\in\mathbb{N}_0}$$

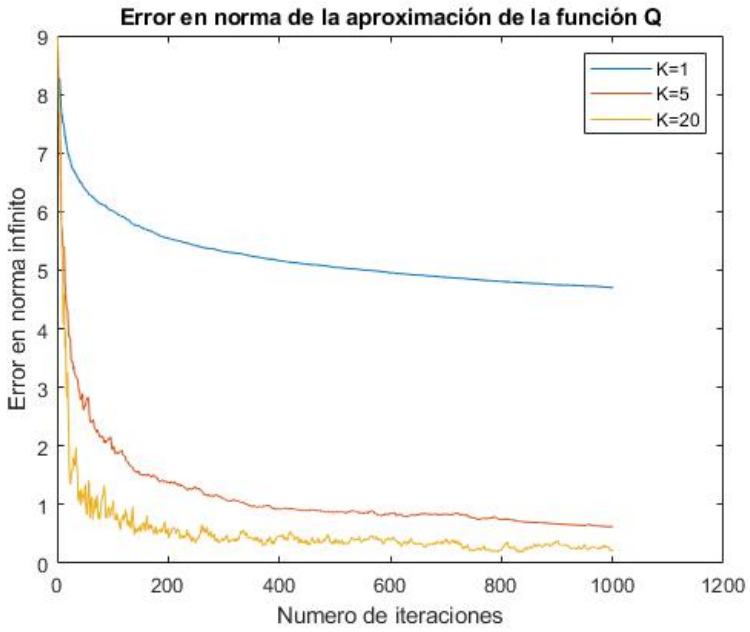


Figura 3.3: Error $\|Q_t - Q^*\|_{\mathcal{Q}}$ como función del número de iteraciones para uno de los procesos de decisión de Markov expuesto en la Sección 2.4 con $\lambda = 0{,}9$. Se muestra para tres sucesiones $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0} = \left\{\frac{1}{1+\lfloor\frac{t}{K}\rfloor}\right\}_{t\in\mathbb{N}_0}$ con $K = 1$, $K = 5$ y $K = 20$. Observamos que el caso $K = 1$ es excesivamente conservador en las iteraciones, mientras que los casos $K = 5$ y $K = 20$ muestran un balance más adecuado entre favorecer la flexibilidad y aceptar una mayor influencia del ruido.

que no decaigan tan rápidamente como $\{\frac{1}{t+1}\}_{t\in\mathbb{N}_0}$, cabría esperar que dejando el resto de parámetros fijos el error decaiga como

$$\epsilon = \mathcal{O}(\sqrt{\frac{1}{T}}).$$

Procediendo de esta manera, el autor tiene motivos para creer que es posible mejorar la expresión (3.10) replicando las ideas de la demostración que se encuentra en [19] para $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0} = \left\{\frac{1}{1+\lfloor\frac{t}{K}\rfloor}\right\}_{t\in\mathbb{N}_0}$ y escogiendo K a posteriori para que el segundo factor domine al primero asintóticamente. Desafortunadamente, por falta de tiempo no ha sido capaz de formalizar con total detalle el argumento teórico; no obstante, como veremos en la próxima sección, al menos el comportamiento empírico parece sugerir que estos razonamientos pueden ser ciertos.

Del mismo modo, utilizando motivaciones similares, se puede intentar argumentar de manera similar para la cota del Teorema 3.3.16. En este caso, *grosso modo*, los dos términos de la cota entre los que había que tomar el máximos se han sumado en vez de multiplicado, pero la intuición del fenómeno subyacente es la misma, y una vez más se corrobora experimentalmente que la sucesión $\{\alpha(s,a)\}_{t\in\mathbb{N}_0} = \{\frac{1}{(t+1)^\beta}\}_{t\in\mathbb{N}_0}$ con

$\beta \in (\frac{1}{2}, 1)$ decae demasiado rápido a efectos prácticos cuando β es próximo a uno. En efecto, la propia expresión (3.11) nos lo indica: si consideramos todos los términos aproximadamente 1 excepto el error ϵ, vemos que

$$T = \mathcal{O}\left(\left(\frac{\ln\left(\frac{1}{\epsilon}\right)}{\epsilon^2}\right)^{\frac{1}{\beta}} + \left(\ln\frac{1}{\epsilon}\right)^{\frac{1}{1-\beta}}\right)$$

Para valores de $\epsilon \sim 10^{-1}$, vemos que el segundo término (relacionado con el decaimiento determinista de los $\{Y_{t;t_k}\}_{t \geq t_k}$) dominará al primero, por lo que el método estaría limitado para este objetivo por su excesiva estabilidad. Por otro lado, se puede llegar a dar también el otro extremo: para valores de β cercanos a $\frac{1}{2}$ y tomando ϵ muy pequeño, la acotación puede venir controlada por el primer término, que indicaría que la limitación viene ahora por influencia del ruido.

Por último, finalizamos esta sección notando que el *rebalanceo* del error que realizan las sucesiones $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0} = \{\frac{1}{(1+t)^\beta}\}_{t\in\mathbb{N}_0}$ al ajustar $\beta \in (\frac{1}{2}, 1)$ no parece ser tan adecuado como el de las sucesiones

$$\left\{\frac{1}{1+\left\lfloor\frac{t}{K}\right\rfloor}\right\}_{t\in\mathbb{N}_0}$$

ni desde el punto de vista *teórico* ni desde la experiencia práctica: en efecto, se comprueba que - como indica (3.11) - con las sucesiones $\{\frac{1}{(1+t)^\beta}\}_{t\in\mathbb{N}_0}$ el error no va a decaer como $\mathcal{O}(\sqrt{\frac{1}{T}})$, mientras que los razonamientos esbozados y las simulaciones numéricas sí que muestran que con $\left\{\frac{1}{1+\left\lfloor\frac{t}{K}\right\rfloor}\right\}_{t\in\mathbb{N}_0}$ el error sigue ese comportamiento. En cierto modo, estas sucesiones parecen combinar las dos cualidades que se requerían: por un lado, en los primeros instantes no decaen excesivamente rápido, pero tras una cantidad significativa de iteraciones sí que recuperan el comportamiento de $\frac{1}{t}$. Son estos los motivos por los que las vamos a emplear en la próxima sección.

3.4. Aplicación numérica

Una vez se ha establecido teóricamente la convergencia del algoritmo y se ha discutido su carácter en muestras finitas, pasamos a presentar algunos resultados numéricos sobre el comportamiento empírico del método. Como ya se comentó, utilizaremos como referencia los resultados expuestos en la Sección 2.4; para los valores de ϵ tomados entonces, se puede comprobar - calculando la función Q asociada a esa aproximación de V y argumentando como en el Teorema 3.2.14 - que efectivamente las políticas ahí obtenidas son óptimas.

Establecido esto, procedemos a retomar en primer lugar el problema del agente sometido a perturbaciones que busca salir de la habitación en la que se encuentra.

Comenzamos estudiando el caso en el que no hay obstáculo, y lo tratamos tanto desde el punto de vista del aprendizaje sincrónico que idealiza el caso de exploración perfecta como desde la lente de una exploración ϵ-avariciosa. A continuación, mostramos cómo el algoritmo es también capaz de lidiar sin dificultades con las perturbaciones que introduce el obstáculo ubicado en el centro de la habitación.

Abordamos primero, pues, el caso sincrónico, para el cuál tomamos el mismo coeficiente de aprendizaje en todos los pares $\alpha_t(s,a) = \alpha(t)$. La discusión realizada en la sección anterior sugirió tomar sucesiones de acuerdo a la forma

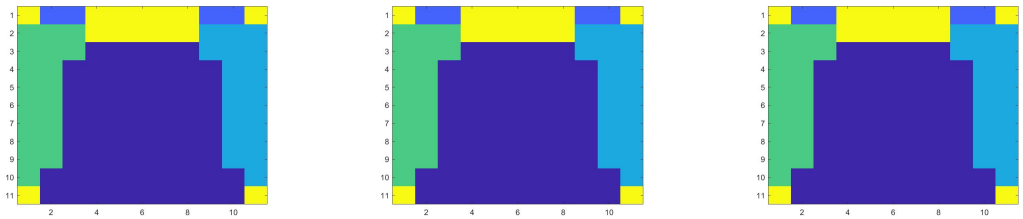
$$\alpha(t) \approx \frac{1}{1 + \lfloor \frac{t}{K} \rfloor}$$

con $K \geq 5$; evidentemente, a mayor K se obtiene más flexibilidad a cambio de estabilidad. La familia de sucesiones así descrita cumple las condiciones $\sum_{t=0}^{\infty} \alpha_t(s,a) = \infty$ y $\sum_{t=0}^{\infty} \alpha_t^2(s,a) < \infty$, por lo que seguimos teniendo garantizada la convergencia del algoritmo. De este modo, una posibilidad válida teóricamente y no patológica empíricamente es considerar para los experimentos la sucesión

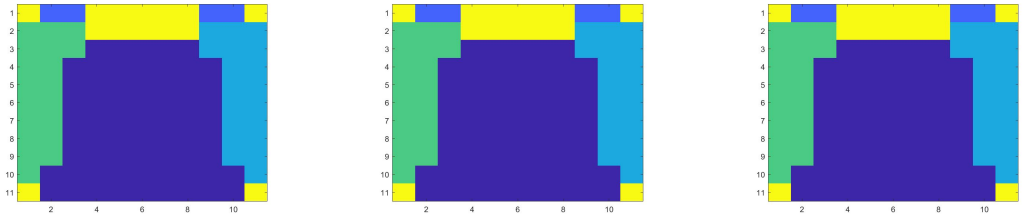
$$\alpha(t) = \frac{1}{2 + \lfloor \frac{t}{10} \rfloor}.$$

Con esto en mente, seguimos la siguiente metodología: para cada valor fijo de λ (al igual que en la Sección 2.4, tomaremos $\lambda \in \{0{,}5, 0{,}7, 0{,}9\}$) consideramos 5 condiciones iniciales diferentes y corremos una aproximación estocástica con $T = 10^5$ para cada una. Puesto que en este problema $M = 1$, se cumple que $\|Q^*\|_{\mathcal{Q}} \leq \frac{1}{1-\lambda}$; así pues, con la finalidad de obtener un error inicial del orden del propio valor a estimar, tomamos los 5 valores de $|Q_0(s,a)|$ aleatoriamente con distribución uniforme entre $[-\frac{1}{1-\lambda}, \frac{1}{1-\lambda}]$. Tras 10^5 iteraciones, extraemos las 5 políticas estacionarias deterministas asociadas a cada estimación de Q^* y las comparamos con la política óptima que se obtuvo tomando el modelo subyacente como conocido y aplicando la iteración de valor. Los resultados obtenidos se encuentran en las Figuras 3.4, 3.5 y 3.6; como se puede observar, el algoritmo estima correctamente la política más adecuada en todos los ensayos menos en dos, donde sugiere políticas muy cercanas pero no idénticas a la óptima.

Con el objeto de comprender más en profundidad el comportamiento del algoritmo, procedemos a ilustrar la evolución del error $\|Q_t - Q^*\|_{\mathcal{Q}}$ para los tres valores de λ y cada uno de los ensayos en la Figura 3.7. Inspeccionando esta gráfica podemos extraer dos conclusiones. En primer lugar, se observa que para los tres valores de λ el decaimiento del error se comporta como $\mathcal{O}(\sqrt{\frac{1}{T}})$, como parcialmente nos sugiere el Teorema 3.3.15; sin embargo, se comprueba que la dependencia del error con el parámetro λ no es tan explosiva como nos dice esa pesimista cota. De hecho, un análisis más cuidadoso nos muestra que el mayor tamaño del error no se debe al deterioro que pueda causar una menor contracción del operador H, sino que tiene su origen en el mayor orden

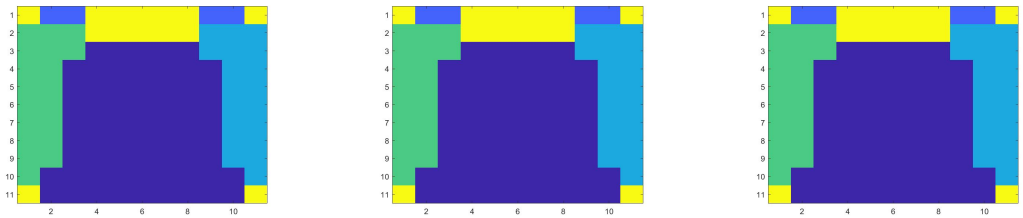


(a) *Value iteration* (b) Test 1 (c) Test 2

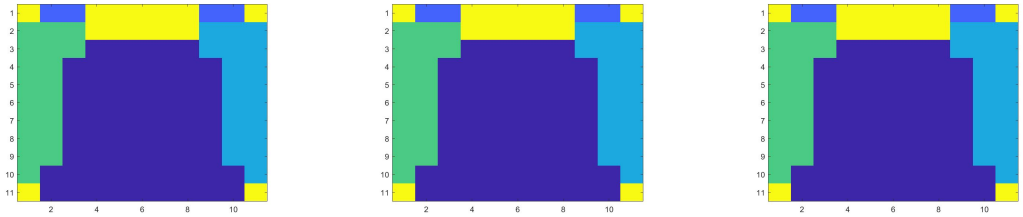


(d) Test 3 (e) Test 4 (f) Test 5

Figura 3.4: Políticas estacionarias obtenidas con el algoritmo de Q-learning para 5 simulaciones distintas con $\lambda = 0{,}5$. Código de colores: azul oscuro $\rightarrow$ arriba, azul $\rightarrow$ abajo, verde $\rightarrow$ derecha, azul claro $\rightarrow$ izquierda, amarillo $\rightarrow$ no hay elección.



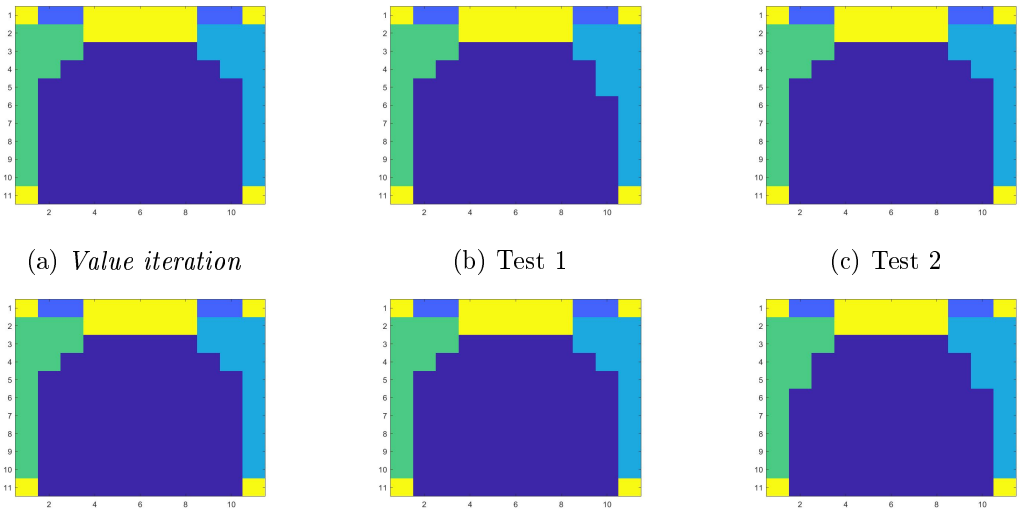
(a) *Value iteration* (b) Test 1 (c) Test 2



(d) Test 3 (e) Test 4 (f) Test 5

Figura 3.5: Políticas estacionarias obtenidas con el algoritmo de Q-learning para 5 simulaciones distintas con $\lambda = 0{,}7$. Código de colores: azul oscuro $\rightarrow$ arriba, azul $\rightarrow$ abajo, verde $\rightarrow$ derecha, azul claro $\rightarrow$ izquierda, amarillo $\rightarrow$ no hay elección.

de magnitud de los números $Q^*(s,a)$ a estimar cuando $\lambda = 0{,}9$ (es decir, el efecto de $V_{\text{máx}}$; obsérvese que las componentes de Q^* ahora serán del orden de $\frac{1}{1-0{,}9} = 10$). Para ver que esto es así, basta normalizar el problema y tomar en cada caso las recompensas $\tilde{r}_\lambda = (1-\lambda)r$, de modo que $\|\tilde{Q}^*\|_{\mathcal{Q}} = \|(1-\lambda)Q^*\|_{\mathcal{Q}} = 1$. En este caso, notando que



(a) *Value iteration* (b) Test 1 (c) Test 2

(d) Test 3 (e) Test 4 (f) Test 5

Figura 3.6: Políticas estacionarias obtenidas con el algoritmo de Q-learning para 5 simulaciones distintas con $\lambda = 0{,}9$. Código de colores: azul oscuro $\rightarrow$ arriba, azul $\rightarrow$ abajo, verde $\rightarrow$ derecha, azul claro $\rightarrow$ izquierda, amarillo $\rightarrow$ no hay elección.

los escalares positivos pueden *entrar* y *salir* del operador no lineal H, es fácil observar que en todo instante t se cumple que

$$\|Q_t - Q^*\|_{\mathcal{Q}} = \frac{1}{1-\lambda}\|\tilde{Q}_t - \tilde{Q}^*\|_{\mathcal{Q}}.$$

Continuando este razonamiento, comprobamos reescalando los datos que efectivamente los problemas normalizados tienen un error que se comporta del mismo modo para los tres valores de λ. De este modo, se concluye que la convergencia en términos de error relativo es similar en las tres situaciones.

A partir de esta observación e inspeccionando los resultados de los ensayos es posible explicar también por qué en el caso de $\lambda = 0{,}9$ no se obtiene en los cinco tests la política óptima exacta. Para este valor de λ, se puede observar que -por ejemplo - la diferencia de $Q^*(s, u_p)$ y $Q^*(s, r_i)$ para los estados de la esquina superior izquierda es en términos relativos del orden de 10^{-3}, mientras que en el caso de $\lambda = 0{,}5$ y $\lambda = 0{,}7$ es del orden de 10^{-2}. Como ya hemos visto que el algoritmo aproxima igual en términos relativos para los tres valores de λ, es natural que entonces las diferencias entre política óptima y estimada solo aparezcan en el caso $\lambda = 0{,}9$. Por tanto, se puede concluir que en el caso $\lambda = 0{,}9$ no es que el algoritmo deje de acertar por inestabilidades asociadas al escaso grado de contracción, si no que se debe a una cuestión intrínseca del propio proceso de decisión de Markov para ese valor de λ: hay varias políticas muy cerca de la política óptima. Obsérvese que es natural que el proceso de decisión de Markov subyacente se complique cuando tomamos valores de λ más altos, pues estamos forzando al agente a considerar en más detalle las consecuencias futuras de sus acciones actuales.

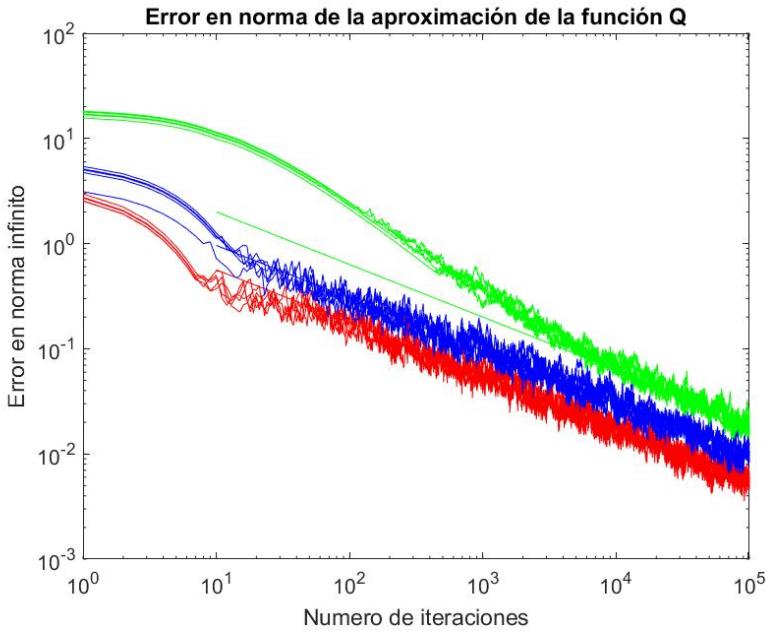


Figura 3.7: Error $\|Q_t - Q^*\|_{\mathcal{Q}}$ como función del número de iteraciones, junto con la recta de pendiente $-\frac{1}{2}$ en escala logarítmica que mejor ajusta de manera conjunta los 5 ensayos para cada valor de λ. Rojo: $\lambda = 0{,}5$; Azul: $\lambda = 0{,}7$; Verde: $\lambda = 0{,}9$.

Para comprobar que ésta es la patología y que efectivamente el algoritmo converge adecuadamente se puede mirar a la magnitud que en realidad nos interesa: el valor asociado a la política estimada, $V_\lambda^{a^\infty_{Q-l}}$, y su error respecto a la cota superior V_λ^*. Recuérdese que, en última instancia, lo que nos interesa es obtener una política los más cercana posible a la óptima en términos de $\|V_\lambda^{a^\infty_{Q-l}} - V_\lambda^*\|_{\mathcal{V}}$, pues esta es la métrica que nos indica lo bien o mal que ha respondido nuestro agente tras el entrenamiento. Los resultados que se obtienen a este respecto se muestran en la Figura 3.8; se observa que, cuando no es directamente 0 (ver, por ejemplo, el final de las líneas rojas, donde faltan puntos), el error es muy pequeño y, una vez más, decrece al mismo ritmo para los tres valores de λ. De nuevo, inspeccionando los resultados se puede volver a comprobar que el mayor error asociado a $\lambda = 0{,}9$ se debe a que los valores de $V^*_{0,9}$ son mayores en módulo (≈ 10) que los de $V^*_{0,7}$ y $V^*_{0,5}$ ($\approx 3{,}33$ y ≈ 2, respectivamente). Concluimos, por tanto, que en términos de error relativo los tres valores de λ se comportan igual. Debe destacarse, eso sí, que estas observaciones, lógicamente, no son extrapolables a cualquier valor de λ: para valores de 0,99 ya se empiezan a observar comportamientos diferentes debido a la escasa contracción que introduce el operador H.

Por último, de la observación de las Figuras 3.7 y 3.8 y con vistas a las aplicaciones, se puede concluir que, por la forma en la que decaen los errores, no es necesario emplear tantas iteraciones como en estos ensayos para obtener una precisión satisfactoria de cara a las aplicaciones. Éste fenómeno se acentúa aún más si cabe en la

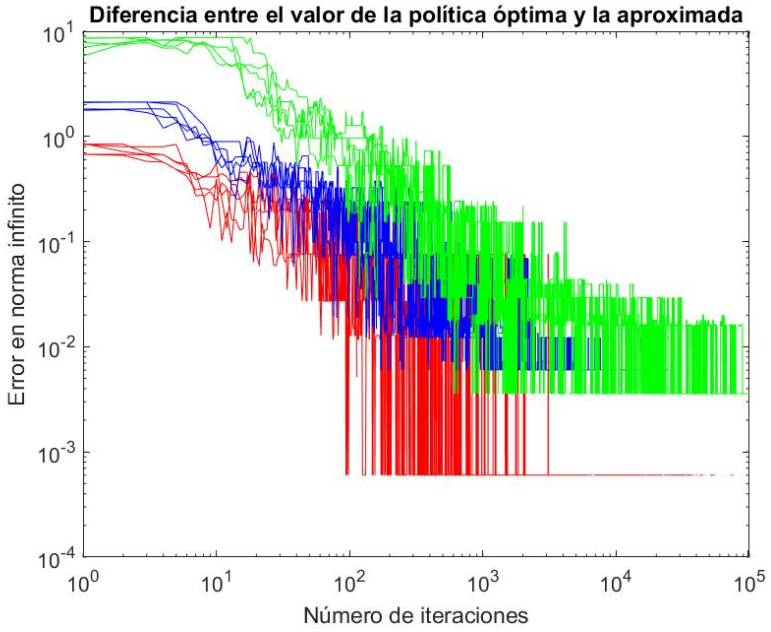


Figura 3.8: Error $\|V_\lambda^{a^\infty_{Q-l}} - V_\lambda^*\|_\nu$ como función del número de iteraciones, siendo a^∞_{Q-l} la política que sugiere el algoritmo en para ese instante del ensayo. Se muestran los resultados de los 5 experimentos para cada valor de λ. Rojo: $\lambda = 0{,}5$; Azul: $\lambda = 0{,}7$; Verde: $\lambda = 0{,}9$.

magnitud que en el fondo nos es relevante, $V_\lambda^{a^\infty_{Q-l}}$.

Habiendo comprobado que efectivamente el algoritmo se comporta de manera adecuada en el caso sincrónico pasamos ahora a estudiar cómo responde el método cuando se utiliza de manera simultánea con una técnica de exploración. En los próximos párrafos, se discuten los resultados de aplicar la iteración de Q-learning junto con una exploración ϵ-avariciosa como la presentada en la Subsección 3.2.3. Los parámetros se seleccionaron de forma que el valor de la *avaricia* $1-\epsilon$ es 0,8 y la sucesión $\{\alpha(t)\}_{t\in\mathbb{N}_0}$ es, de nuevo,

$$\alpha(t) = \frac{1}{2 + \lfloor \frac{t}{10} \rfloor}.$$

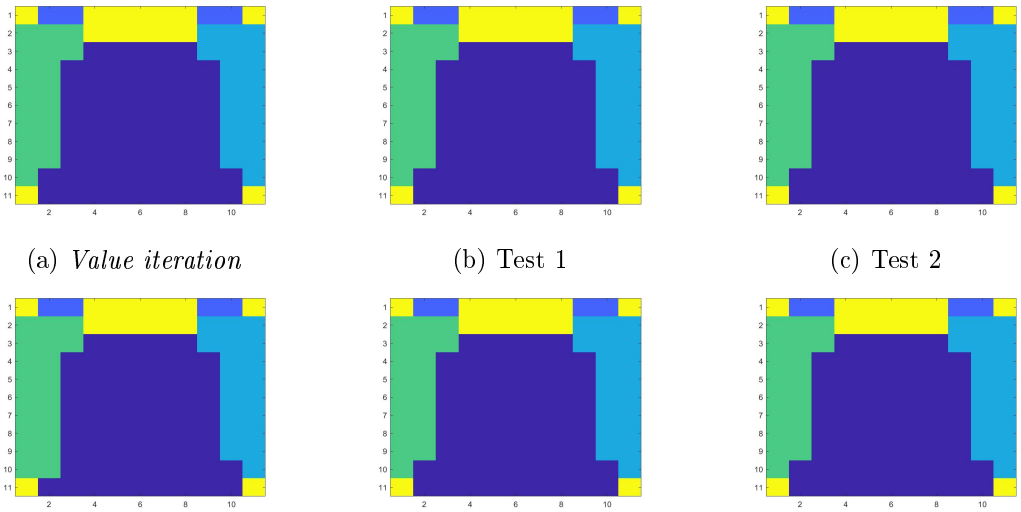
Recordamos que, bajo el paradigma de la exploración ϵ-avariciosa, el valor del coeficiente de aprendizaje $\alpha_t(s,a)$ viene dado por

$$\alpha_t(s,a) = \alpha(\#[s,a,t]),$$

donde $\#[s,a,t]$ es el número de veces que se ha visitado el par estado-acción (s,a) en lo que va de realización de algoritmo antes del instante t. Por último, para concluir la discusión sobre el ajuste de parámetros, se fijó para estos ensayos la cantidad de pasos L hasta que se reubica el agente de manera aleatoria como $L = N = 11$, y el número de iteraciones $T = 11 \cdot 10^6$; por tanto, en cada realización se simulan 10^6 cadenas de acciones de longitud $L = 11$. El motivo por el que se tomó este valor de

T fue el de equiparar el coste computacional de este enfoque con el del ejemplo anterior. Obsérvese que tanto antes como ahora se producen aproximadamente el mismo número de actualizaciones: en el caso anterior, $10^5 \cdot 11 \cdot 11$ (10^5 iteraciones con $11 \cdot 11$ actualizaciones en cada paso temporal), mientras que en el actual $\cdot 10^6 \cdot 11$ ($10^6 \cdot 11$ iteraciones con una actualización en cada paso temporal).

Al igual que en el caso anterior, se corrieron 5 experimentos para cada valor de $\lambda \in \{0{,}5, 0{,}7, 0{,}9\}$ con condiciones iniciales distribuidas uniformemente en el intervalo $[-\frac{1}{1-\lambda}, \frac{1}{1-\lambda}]$. De nuevo, como se muestra en las Figuras 3.9, 3.10 y 3.11, el algoritmo obtiene en la mayoría de los casos la política óptima; el caso en el que a veces estima la política con pequeñas modificación es el de $\lambda = 0{,}9$, en el que - como ya se comentó - la tolerancia relativa necesaria es particularmente pequeña debido a las múltiples políticas con valor próximo a la óptima. En lenguaje de optimización, la dificultad radica en que ese máximo es muy plano.

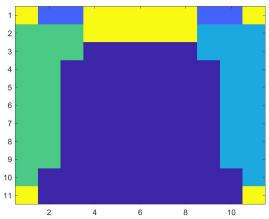


(a) *Value iteration* (b) Test 1 (c) Test 2

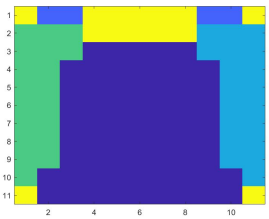
(d) Test 3 (e) Test 4 (f) Test 5

Figura 3.9: Políticas estacionarias obtenidas con la iteración de Q-learning y exploración ϵ-avariciosa para 5 simulaciones distintas con $\lambda = 0{,}5$. Código de colores: azul oscuro $\rightarrow$ arriba, azul $\rightarrow$ abajo, verde $\rightarrow$ derecha, azul claro $\rightarrow$ izquierda, amarillo $\rightarrow$ no hay elección.

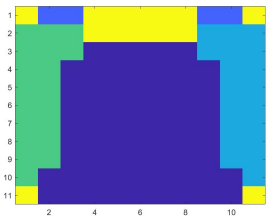
Como ya se comentó, estos algoritmos asíncronos padecen de la imposibilidad de incorporar a la vez toda la información e interrelaciones entre los diferentes estados, por lo que buscan compensar esa desventaja con una organización mas *eficiente* de las visitas a los pares estado-acción $(s, a) \in S \times A_S$. Por el carácter ϵ-avaricioso de la exploración, es de esperar que se visiten más los pares (s, a) más favorables, y por la metodología de simular en los estados en cadena, cabe esperar que se visiten más veces estados en los que confluyen muchas secuencias como los próximos a la puerta. Como se muestra en la Figura 3.12, se observa que efectivamente el comportamiento



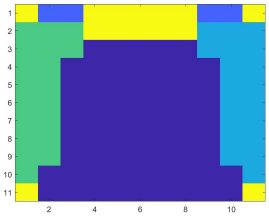
(a) *Value iteration*



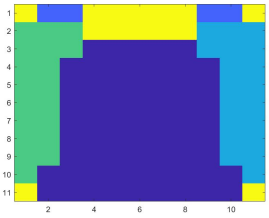
(b) Test 1



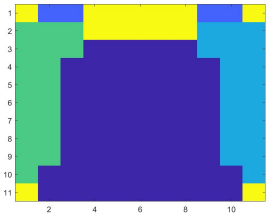
(c) Test 2



(d) Test 3

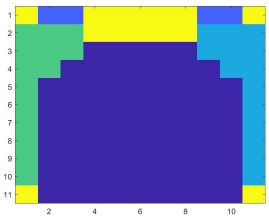


(e) Test 4

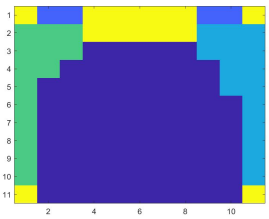


(f) Test 5

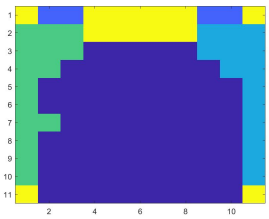
Figura 3.10: Políticas estacionarias obtenidas con la iteración de Q-learning y exploración ϵ-avariciosa para 5 simulaciones distintas con $\lambda = 0{,}7$. Código de colores: azul oscuro $\rightarrow$ arriba, azul $\rightarrow$ abajo, verde $\rightarrow$ derecha, azul claro $\rightarrow$ izquierda, amarillo $\rightarrow$ no hay elección.



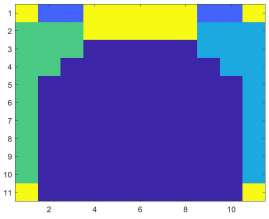
(a) *Value iteration*



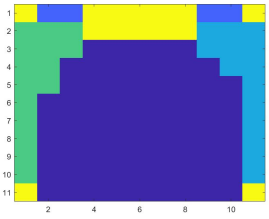
(b) Test 1



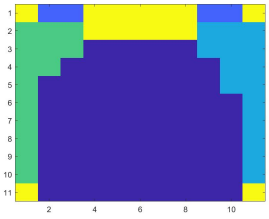
(c) Test 2



(d) Test 3



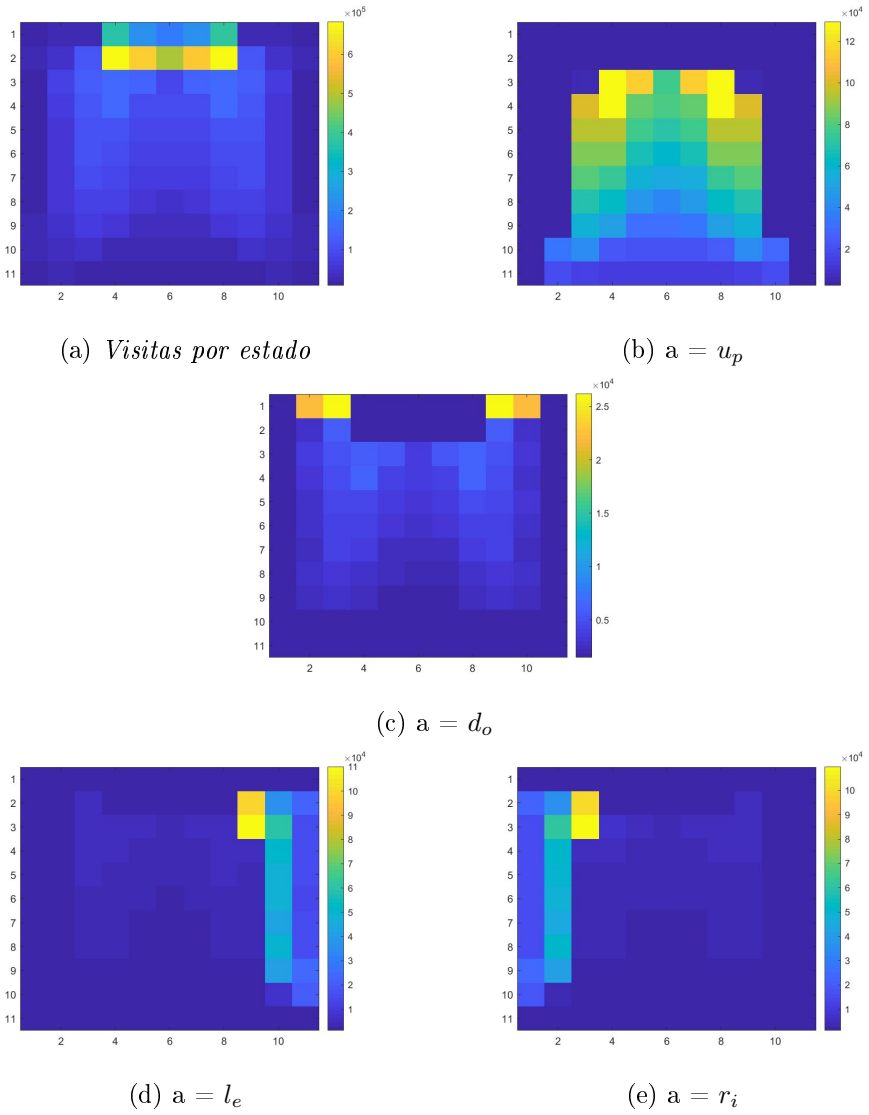
(e) Test 4



(f) Test 5

Figura 3.11: Políticas estacionarias obtenidas con la iteración de Q-learning y exploración ϵ-avariciosa para 5 simulaciones distintas con $\lambda = 0{,}9$. Código de colores: azul oscuro $\rightarrow$ arriba, azul $\rightarrow$ abajo, verde $\rightarrow$ derecha, azul claro $\rightarrow$ izquierda, amarillo $\rightarrow$ no hay elección.

de las visitas es el previsto.



(a) *Visitas por estado* (b) a = u_p

(c) a = d_o

(d) a = l_e (e) a = r_i

Figura 3.12: Mapa de calor del número de visitas realizado. En primer lugar, se muestra las frecuencias del número de visitas por estado. Las siguientes figuras muestran la distribución de visitas a los pares (s, a) con a fijo.

Por último, con el fin de cuantificar la precisión que se consigue con la combinación de la iteración de Q-learning y la exploración ϵ-avariciosa se podría estudiar la evolución del error $\|Q_t - Q^*\|_{\mathcal{Q}}$. Sin embargo, se puede razonar que en realidad esta no es una métrica justa: si bien el toque de aleatoriedad tanto en la reubicación de estados como en la elección de acciones me garantiza que esa norma ha de converger a cero, lo que en realidad está buscando el algoritmo es priorizar la actualización de los pares (s, a) que satisfacen $Q^*(s, a) = \max_{b \in A_s} Q^*(s, b)$. De este modo, dado que la exploración se concentra en los valores $V_t(s) = \max_{b \in A_s} Q_t(s, b)$, una métrica del error más adecuada podría ser, por ejemplo, $\| \max_{b \in A_s} Q_t(s, b) - \max_{b \in A_s} Q^*(s, b)\|_{\mathcal{V}} = \|V_t(s) - V^*(s)\|_{\mathcal{V}}$. Éste es el resultado que se presenta en la Figura 3.13, donde se observa que en esta

magnitud el error alcanzado es similar al del caso sincrónico. Una vez más, argumentando como anteriormente, el mayor error absoluto que se observa para un valor de λ más grande no traduce en un mayor error relativo.

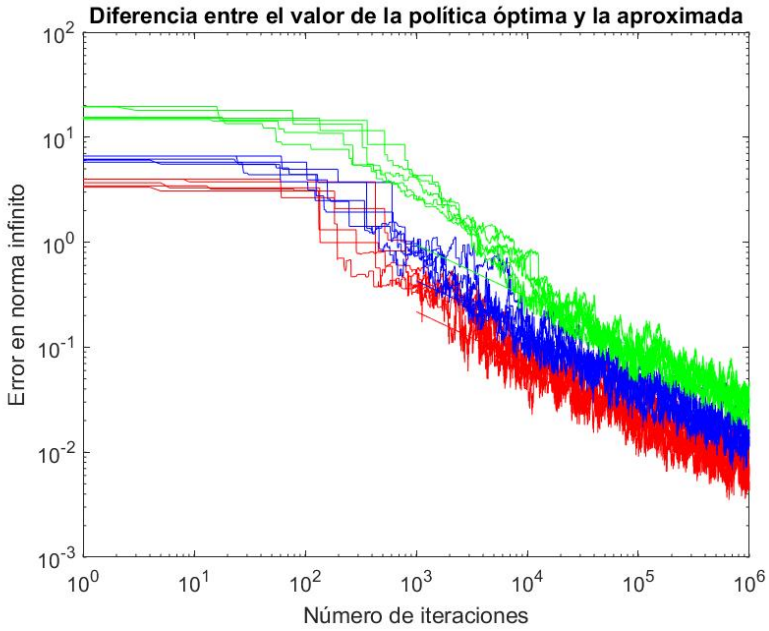


Figura 3.13: Error $\| \max_{b \in A_s} Q_t(s,b) - \max_{b \in A_s} Q^*(s,b) \|_{\mathcal{V}}$ como función del número de iteraciones para el algoritmo de Q-learning junto con una exploración ϵ-avariciosa. Se muestran los resultados de los 5 experimentos para cada valor de λ, junto con la recta de pendiente $-\frac{1}{2}$ que mejor los ajusta conjuntamente. Rojo: $\lambda = 0{,}5$; Azul: $\lambda = 0{,}7$; Verde: $\lambda = 0{,}9$.

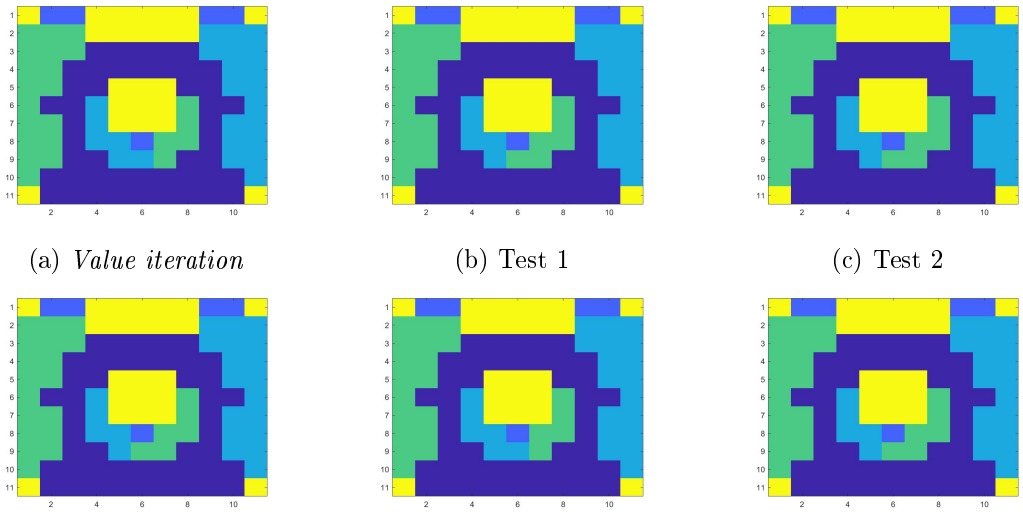
Concluimos el estudio del problema de la habitación abordando con el algoritmo de Q-learning el caso ya presentado en la Sección 2.4 en el que se coloca un obstáculo en el centro de la estancia. Nos restringimos al caso sincrónico y, con el objeto de establecer similitudes con los resultados expuestos hasta ahora, tomamos los mismos parámetros: $T = 10^5$, $N = 11$ y $\{\alpha_t(s,a)\} = \{\alpha(t)\}_{t \in N_0}$ de modo que

$$\alpha(t) = \frac{1}{2 + \lfloor \frac{t}{10} \rfloor}.$$

Asímismo, seguimos la misma metodología: consideramos $\lambda \in \{0{,}5, 0{,}7, 0{,}9\}$, y para cada valor de estos corremos cinco veces el algoritmo con condiciones iniciales aleatorias.

Comenzamos analizando los resultados comparando las políticas sugeridas por Q-learning con las óptimas en las Figuras 3.14, 3.15 y 3.16. Como se puede observar, la introducción del obstáculo no supone un problema para el algoritmo: a excepción de un estado en el que claramente hay multiplicidad de acciones óptimas, en la zona en las que las decisiones se ven afectadas por el nuevo muro la estimación de la política

coincide exactamente con la óptima en todas las realizaciones. De hecho, en cierto modo, la política en esa zona es más sencilla de estimar que en las esquinas superiores, donde sigue apareciendo la ligera imprecisión que se observaba antes para el caso $\lambda = 0{,}9$: esto se debe a que la distorsión introducida por el obstáculo marca de forma más abrupta cuál es la acción más adecuada.



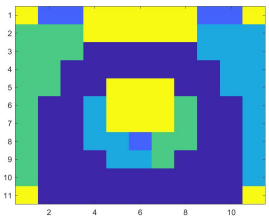
(a) *Value iteration* (b) Test 1 (c) Test 2

(d) Test 3 (e) Test 4 (f) Test 5

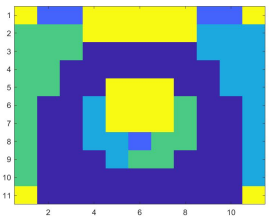
Figura 3.14: Políticas estacionarias obtenidas con la iteración de Q-learning en el problema con obstáculo para 5 simulaciones distintas con $\lambda = 0{,}5$. Código de colores: azul oscuro → arriba, azul → abajo, verde → derecha, azul claro → izquierda, amarillo → no hay elección.

Para entender en más profundidad este fenómeno, se puede observar los resultados de las Figuras 3.17 y 3.18. En la primera de ellas, se muestra el comportamiento del error $\|Q_t - Q^*\|_{\mathcal{Q}}$ como función del número de iteraciones y es posible extraer dos conclusiones al respecto. En primer lugar, notamos que, de nuevo, para los tres valores de λ se observa que el error decae como $\mathcal{O}(\sqrt{\frac{1}{T}})$. Además, observamos otra vez el fenómeno discutido anteriormente: el error absoluto es mayor para valores mayores de λ, pero es fácil comprobar mediante reescalados que el error relativo es similar en los tres escenarios.

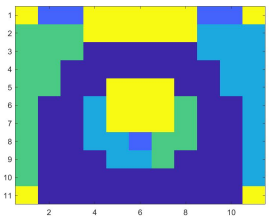
En segundo lugar, se observa que ahora los valores del error en términos absolutos son ligeramente mayores que los que se ilustraban en la Figura 3.7; este fenómeno, por otra parte, no es sorprendente, pues es claro que la función Q^*_λ varía ahora de forma más abrupta y por tanto es más difícil de estimar. No obstante, estas circunstancias en las que hay variaciones fuertes en los valores de Q^*_λ son, a la postre, positivas para el algoritmo: en estas situaciones los números $Q^*(s, a)$ para s fijo toman valores muy distintos por lo que, aunque la estimación sea ligeramente menos precisa, no hace falta tanta exactitud para determinar adecuadamente cuál es la acción más adecuada. En



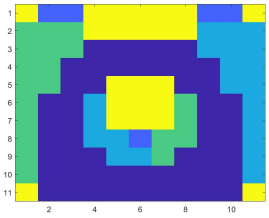
(a) *Value iteration*



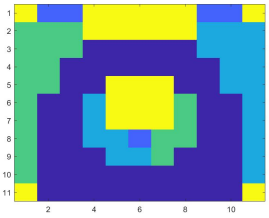
(b) Test 1



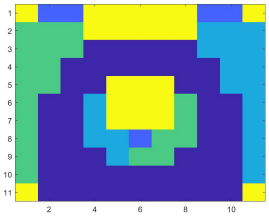
(c) Test 2



(d) Test 3

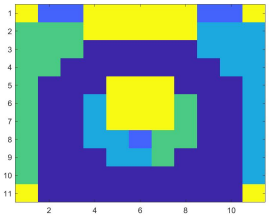


(e) Test 4

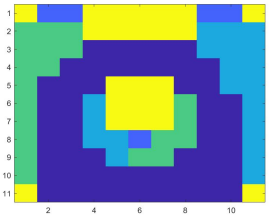


(f) Test 5

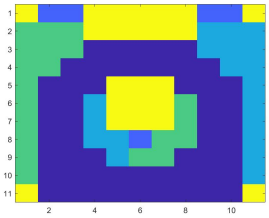
Figura 3.15: Políticas estacionarias obtenidas con la iteración de Q-learning en el problema con obstáculo para 5 simulaciones distintas con $\lambda = 0{,}7$. Código de colores: azul oscuro → arriba, azul → abajo, verde → derecha, azul claro → izquierda, amarillo → no hay elección.



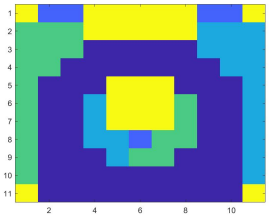
(a) *Value iteration*



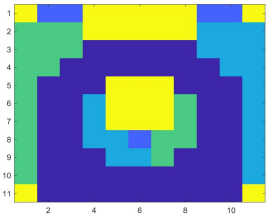
(b) Test 1



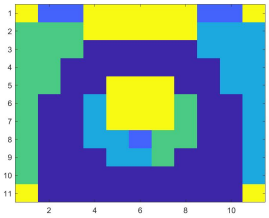
(c) Test 2



(d) Test 3



(e) Test 4



(f) Test 5

Figura 3.16: Políticas estacionarias obtenidas con la iteración de Q-learning en el problema con obstáculo para 5 simulaciones distintas con $\lambda = 0{,}9$. Código de colores: azul oscuro → arriba, azul → abajo, verde → derecha, azul claro → izquierda, amarillo → no hay elección.

efecto, y como se muestra en la Figura 3.18, el algoritmo es ahora más rápido a la hora de acercar la magnitud que nos es verdaderamente relevante, $V_\lambda^{a^\infty_{Q-l}}$, a la cota

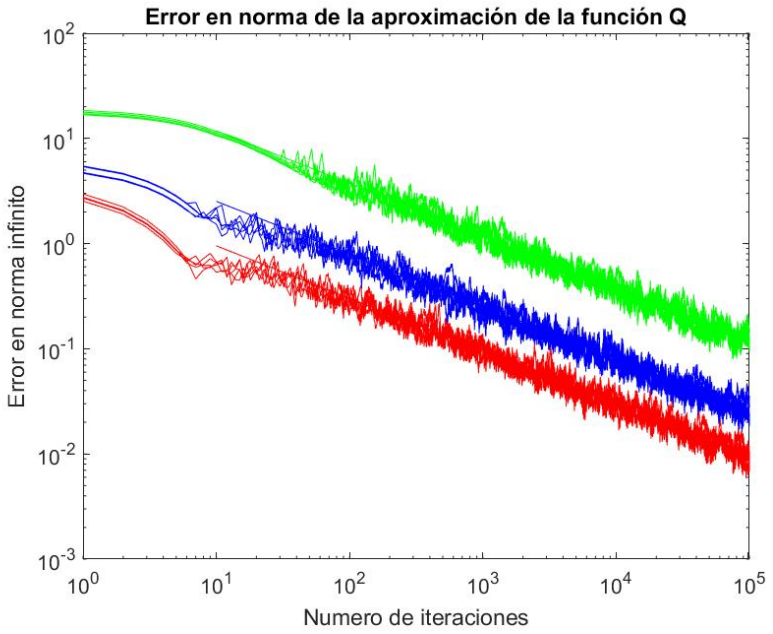


Figura 3.17: Error $\|Q_{\lambda t} - Q^*_\lambda\|_{\mathcal{Q}}$ como función del número de iteraciones para el problema con obstáculo, junto con la recta de pendiente $-\frac{1}{2}$ en escala logarítmica que mejor ajusta de manera conjunta los 5 ensayos para cada valor de λ. Rojo: $\lambda = 0{,}5$; Azul: $\lambda = 0{,}7$; Verde: $\lambda = 0{,}9$.

superior V^*_λ. De hecho, como muestran las zonas en las que no hay puntos, el error se hace nulo en muchos instantes temporales a partir de las 10^4 iteraciones.

Con la finalidad de comprobar que las impresiones extraídas de los resultados expuestos hasta ahora gozan de cierta generalidad procedemos a examinar ahora el comportamiento de magnitudes similares en el problema de los proyectiles descrito en la Sección 2.4. Como ya se comentó, la tridimensionalidad del espacio de estados complica la visualización de las políticas estimadas, así que en este ejemplo solo será posible evaluar el comportamiento del algoritmo examinando magnitudes como los errores $\|Q_t - Q^*\|_{\mathcal{Q}}$ o $\|V_\lambda^{a^\infty_{Q-l}} - V^*_\lambda\|_{\mathcal{V}}$.

La metodología empleada para el análsis experimental de este procesos de decisión de Markov vuelve a ser la misma que en los estudios anteriores: se fijaron varios valores de λ ($\lambda \in \{0{,}5, 0{,}7, 0{,}9\}$) y para cada uno de ellos se realizaron 5 ensayos con condiciones iniciales tomadas de manera aleatoria uniformemente distribuida en $[-\frac{1}{1-\lambda}, \frac{1}{1-\lambda}]$. Con el objeto de no extender demasiado la exposición, para este proceso solo se consideró el caso de aprendizaje sincrónico, con $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0} = \{\alpha(t)\}_{t\in\mathbb{N}_0}$ siendo, de nuevo,

$$\alpha(t) = \frac{1}{2 + \lfloor \frac{t}{10} \rfloor}.$$

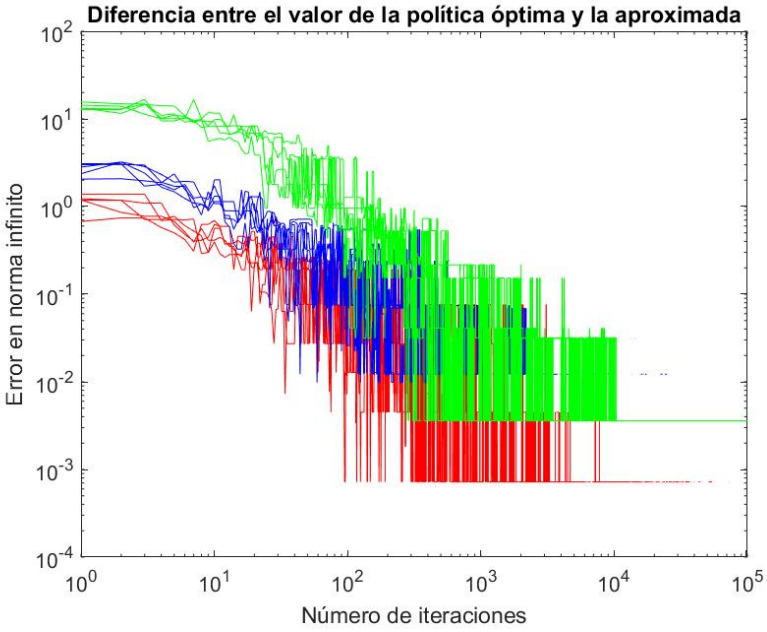


Figura 3.18: Error $\|V_\lambda^{a^\infty_{Q-l}} - V_\lambda^*\|_\nu$ como función del número de iteraciones, siendo a^∞_{Q-l} la política que sugiere el algoritmo en para ese instante del ensayo. Se muestran los resultados de los 5 experimentos sobre el problema con obstáculo para cada valor de λ. Rojo: $\lambda = 0{,}5$; Azul: $\lambda = 0{,}7$; Verde: $\lambda = 0{,}9$.

En las Figuras 3.19 y 3.20 se muestran los resultados obtenidos, en los que se observa comportamientos ya discutidos como el decaimiento $\mathcal{O}(\sqrt{\frac{1}{T}})$ del error o el mayor tamaño del error absoluto (¡no necesariamente el relativo!) para λ mayor. Se concluye, pues, que para este problema el comportamiento del algoritmo es igual de efectivo o incluso más que para el anterior.

De entre las cosas que mejoran en la aplicación de Q-learning a este problema concreto, destaca la rapidez con la que se estima la política óptima: como se comprueba en la Figura 3.20, ésta se alcanza con un número de iteraciones de alrededor de 5000. El motivo por el que el algoritmo goza ahora de más rapidez no está relacionado con la dificultad o simplicidad de la tarea subyacente, pues nótese que tenemos un número de estados similar al anterior (5^3 vs. 11^2) y ahora se observa no solo una si no dos fuentes de ruido (dónde cae el proyectil de la fila superior sumado a dónde se coloca el siguiente). La razón que en realidad causa ese comportamiento mejorado es el mejor planteamiento del problema de optimización subyacente: mientras que antes había varias políticas muy cerca de la óptima, ahora los máximos se encuentran mejor diferenciados. En efecto, como muestra la Figura 3.20, de un error de aproximadamente 10^{-1} se pasa directamente a un error despreciable.

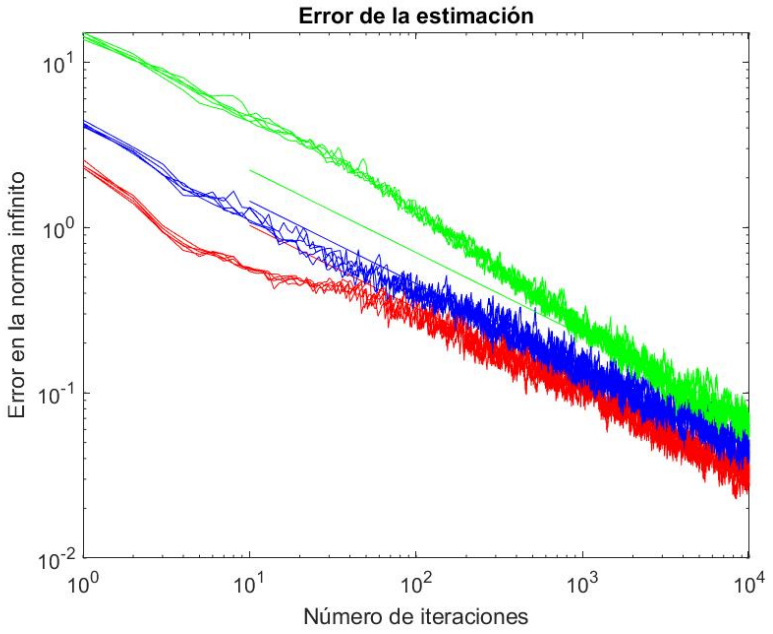


Figura 3.19: Error $\|Q_t - Q^*\|_\nu$ como función del número de iteraciones, junto con la recta de pendiente $-\frac{1}{2}$ en escala logarítmica que mejor ajusta de manera conjunta los 5 ensayos para cada valor de λ. Se muestran los resultados de los 5 experimentos sobre el problema del proyectil para cada valor de λ. Rojo: $\lambda = 0{,}5$; Azul: $\lambda = 0{,}7$; Verde: $\lambda = 0{,}9$.

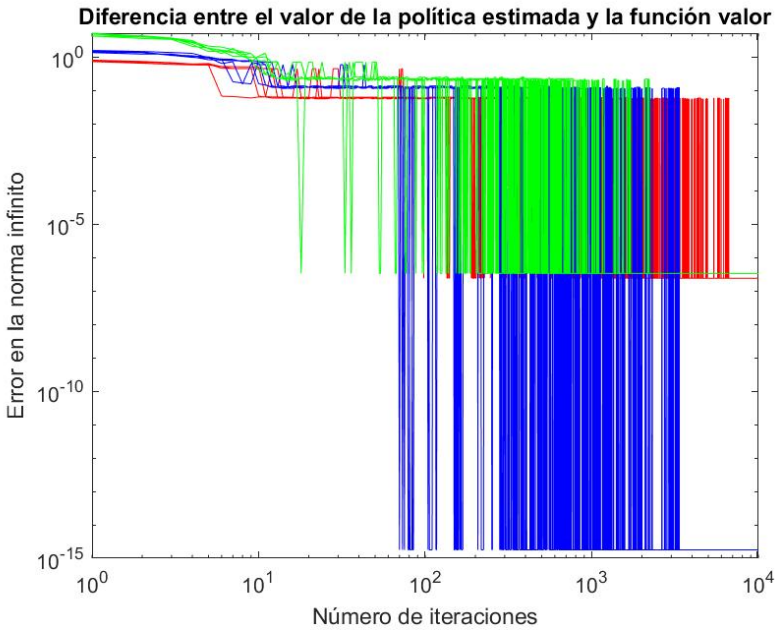


Figura 3.20: Error $\|V_\lambda^{a^\infty_{Q-l}} - V_\lambda^*\|_\nu$ como función del número de iteraciones, siendo a^∞_{Q-l} la política que sugiere el algoritmo en para ese instante del ensayo. Se muestran los resultados de los 5 experimentos sobre el problema del proyectil para cada valor de λ. Rojo: $\lambda = 0{,}5$; Azul: $\lambda = 0{,}7$; Verde: $\lambda = 0{,}9$. Por la tolerancia escogida para los cálculos, a partir de 10^{-6} el error se considera despreciable.

CAPÍTULO 4

Conclusiones y perspectivas futuras

A lo largo de este trabajo se ha propuesto un método para estimar de manera directa la función Q^* de un proceso de decisión de Markov, a partir de la cuál se ha mostrado que es posible sintetizar una política óptima. De la metodología propuesta destaca la flexibilidad que ofrece: con apenas una línea de código,

$$Q_{t+1}(s,a) = (1-\alpha_t(s,a))Q_t(s,a)+\alpha_t(s,a)\Big[r(s,a,s'_t(s,a))+\lambda \max_{b\in A_{s'_t(s,a)}} Q_t(s'_t(s,a),b)\Big],$$

se puede establecer un principio general que permite encontrar la ley de decisión más adecuada en todo instante y estado. El algoritmo de Q-learning tiene garantizada teóricamente su convergencia en sus múltiples versiones (ya sea la sincrónica o la asíncrona combinada con un método de exploración válido) con la única restricción práctica de que la dinámica subyacente a la evolución del agente sea un proceso de decisión de Markov. Afortunadamente, estos procesos permiten modelar una clase muy general de sistemas, por lo que esta hipótesis no es excesivamente restrictiva: los ejemplos expuestos en esta memoria - en los que un agente busca salir de una habitación con obstáculos o trata de evitar unos proyectiles que se mueven con cierta aleatoriedad - no son más que una minúscula muestra del tipo de problemas que se pueden abordar con este marco de trabajo.

Como se ha ilustrado, el algoritmo es capaz de tratar sin problema y con un coste computacional relativamente asequible los problemas aquí considerados. No obstante, en opinión del autor, las garantías sobre el comportamiento práctico para el caso general no son suficientes para considerar el tratamiento matemático del algoritmo totalmente cerrado, ni siquiera en el caso sincrónico. Como ya se discutió en la Sección 3.3, las cotas de las que se dispone para el comportamiento en tiempo finito del algoritmo sincrónico con ciertas sucesiones $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ son mejorables desde un punto de vista tanto técnico como fundamental, y no nos permiten establecer el tamaño de los coeficientes $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ de una forma sistemática para ajustar el comportamiento a las múltiples situaciones de la práctica. El autor tiene motivos (tanto analíticos como numéricos) para creer que los argumentos esbozados en la Sección 3.3 pueden llevar a la obtención tanto de unas mejores estimaciones teóricas como a un criterio para adaptar la sucesión $\{\alpha_t(s,a)\}_{t\in\mathbb{N}_0}$ utilizando otros parámetros conocidos *a priori*. Se

trata este, pues, de un frente interesante de cara a análisis futuros. Cabe destacar que, conceptualmente, esta problemática no es nueva en el entorno de las aproximaciones estocásticas (ver [34]); no obstante, al menos de acuerdo al mejor conocimiento del autor, no ha sido abordada para la estructura particular de la iteración de Q-learning.

Relacionadas con la discusión del ritmo de convergencia en tiempo finito se encuentran las diferentes modificaciones propuestas al algoritmo original. Puesto que el comportamiento en el límite no supone problema para el Q-learning aquí expuesto, el objetivo de estas sugerencias (ver, por ejemplo, [1], [13], [28], [35]) es disminuir el número de iteraciones necesario para obtener una precisión dada con una cierta probabilidad. No obstante, para que las comparaciones sean adecuadas es exigible que las estimaciones que se vayan a utilizar sean razonablemente representativas del comportamiento del algoritmo, por lo que esta segunda línea de trabajo futuro no es sustitutiva sino complementaria de la anterior. Además, debe tenerse en cuenta que algunos de estos algoritmos pueden no disfrutar de la misma generalidad que el original, por lo que quizás solo puedan ser alternativas en ciertas ocasiones.

Otro aspecto interesante de cara a profundizar en su análisis matemático es el del comportamiento a tiempo finito del algortimo utilizado con distintas políticas exploratorias. No obstante, éste es un frente particularmente difícil de abordar por dos motivos. En primer lugar, si ni siquiera está adecuadamente resuelto el caso más sencillo del algoritmo sincrónico no parece sensato tratar de estudiar una versión más complicada del problema. En segundo lugar, como ya se comentó, es difícil de abordar matemáticamente este problema de manera coherente: la exploración se aplica cuando se desconoce el sistema, por lo que es de esperar que las hipótesis sobre las que descansen estos análisis no se puedan verificar en la práctica.

Por último, y como mera referencia a largo plazo, un posible objeto de futuro estudio matemático es la interacción de algoritmos de la familia Q-learning con las técnicas del campo del aprendizaje profundo, dando lugar a lo que se conoce como *Deep Q-learning* y que se encuentra en el corazón de algunas demostraciones tecnológicas sorprendentes (ver el Capítulo 1). De manera resumida, la intersección de ambos campos surge de que en estos enfoques no se trata de aproximar la tabla Q^* utilizando tablas Q_t, sino que se aproxima la tabla Q^* usando redes neuronales. La motivación de esta metodología es trabajar de manera más eficiente problemas con un gran número de estados. No obstante, el tratamiento matemático riguroso de estas técnicas se anticipa complicado, pues requiere estudiar las interacciones de dos métodos que individualmente no están adecuadamente comprendidos.

En cualquier caso, como se puede ver, las técnicas presentadas en este trabajo no solo suponen un logro meritorio de estudio por su propia cuenta, sino que además sirven como puerta de entrada a una gran cantidad de interrogantes matemáticos. El objeto último de todos éstos es el mismo: aportar soporte lógico a áreas muy relevantes de la inteligencia artificial, una de las gran revoluciones científicas y tecnológicas de nuestro tiempo.

APÉNDICE A

Sobre las distribuciones iniciales en los procesos de decisión de Markov

El proceso estocástico inducido en un proceso de decisión de Markov queda - como ya se ha comentado - totalmente determinado por una distribución inicial ν y una política π, que al ser seleccionados determinan la probabilidad $\mathbb{P}_\nu^\pi$. No obstante, al igual que ocurría en las cadenas de Markov, para tratar el caso general no es necesario estudiar el comportamiento de $\mathbb{P}_\nu^\pi$ para leyes iniciales arbitrarias; basta considerar los casos $\nu = \delta_s$ y echar mano de la siguiente igualdad

$$\mathbb{P}_\nu^\pi(B) = \sum_{s\in S} \nu(s)\mathbb{P}_s^\pi(B).$$

El argumento para establecer este hecho es sencillo; las premedidas $\mathbb{P}_\nu^\pi(\cdot)$ y $\tilde{\mathbb{P}}_\nu^\pi(\cdot) = \sum_{s\in S} \nu(s)\mathbb{P}_s^\pi(B)$ coinciden en el álgebra $\mathcal{C}$, pues

$$\begin{aligned}\mathbb{P}_\nu^\pi(C_n^s(s_0,a_0,...,s_n)) &= \nu(s_0)q_1(a_0|h_0)p(s_1|s_0,a_0)\cdots p(s_n|s_{n-1},a_{n-1}) = \\ &= \nu(s_0)\mathbb{P}_{s_0}^\pi(C_n^s(s_0,a_0,...,s_n)) = \\ &= \sum_{s\in S}\nu(s)\mathbb{P}_s^\pi(C_n^s(s_0,a_0,...,s_n)) = \\ &= \tilde{\mathbb{P}}_\nu^\pi(C_n^s(s_0,a_0,...,s_n))\end{aligned}$$

y análogamente para los conjuntos de $\mathcal{C}$ del tipo $C_n^a(s_0,a_0,...,a_n)$. Dado que solo hay una posible extensión a $\mathcal{P}(\Omega)$ para las premedidas de probabilidad, concluimos que $\mathbb{P}_\nu^\pi(\cdot) = \tilde{\mathbb{P}}_\nu^\pi(\cdot)$ como medidas sobre $(\Omega, \mathcal{P}(\Omega))$

En base a esta simple relación se explica que solo nos vayamos a centrar en encontrar políticas que maximicen las cantidades $\mathbb{E}_s^\pi[W(\omega)]$ para cada $s \in S$ (donde W es una determinada variable aleatoria acotada para garantizar que $W \in L^1(\Omega, \mathcal{P}(\Omega), \mathbb{P}_\nu^\pi)$ para toda $\pi \in \Pi$). A partir del comentario anterior, es directo observar (mirándolo primero para funciones simples y tomando límite) que

$$\mathbb{E}_\nu^\pi[W(\omega)] = \sum_{s\in S}\nu(s)\mathbb{E}_s^\pi[W(\omega)].$$

Por tanto, si somos capaces de encontrar una π que maximice $\mathbb{E}_s^\pi [W(\omega)]$ en todo $s \in S$ estaremos implícitamente maximizando $\mathbb{E}_\nu^\pi [W(\omega)]$ para ν arbitraria.

APÉNDICE B

Resultados auxiliares para el análisis de las aproximaciones estocásticas

En este apéndice introducimos un resultado angular de cara a la utilización de las aproximaciones estocásticas en este trabajo. Nuestro objetivo es abordar el comportamiento de una de las aproximaciones estocásticas más sencillas que se pueden pensar,

$$r_{t+1} = (1 - \alpha_t) r_t + \alpha_t w_t, \tag{B.1}$$

donde w_t es una fuente de ruido de tal modo que las variables aleatorias $\{w_t\}_{t \in \mathbb{N}_0}$ son independientes e idénticamente distribuidas con media nula y varianza finita. Intuitivamente vemos que en cada instante temporal se acerca el valor r_t a 0 y se le suma un término asociado a perturbaciones; por tanto, es de esperar que si el sistema converge a algún valor éste habrá de ser necesariamente 0.

Lo primero que observamos heurísticamente es que los valores de $\{\alpha_t\}_{t \in \mathbb{N}}$ han de cumplir unas ciertas condiciones si queremos que (B.1) converja con probabilidad 1 a un cierto valor independientemente de la condición inicial. Un ejemplo de las relaciones que cabe esperar es que se satisfaga que $S_\alpha = \sum_{t=0}^{\infty} \alpha_t = \infty$ para que sea posible alcanzar cualquier valor desde una condición inicial r_0. Pensemos en el siguiente argumento informal: si $|w_t| \leq M$, entonces $|r_t| \leq |r_0| + M \sum_{\tau=0}^{t-1} \alpha_t \leq |r_0| + M S_\alpha$. De este modo,

$$|r_t - r_0| \leq \sum_{\tau=0}^{t-1} |r_{\tau+1} - r_\tau| \leq \sum_{\tau=0}^{t-1} \alpha_t (|r_0| + M S_\alpha + M) \leq |r_0| S_\alpha + M S_\alpha^2 + M S_\alpha.$$

Así pues, para cada valor de $|r_0|$ vemos que es posible encontrar un valor finito de S_α de tal modo que nuestra sucesión verifique que $|r_t - r_0| < |r_0|$ y que, por ende, sea incapaz de converger a 0. Vemos, por tanto, que es adecuado pedir $\sum_{t=0}^{\infty} \alpha_t = \infty$ para no limitar el conjunto de puntos alcanzable desde un valor inicial arbitrario de r_0.

Por otro lado, es evidente que es necesario tener $\lim_{t\to\infty} \alpha_t = 0$, pues si no la varianza de r_{t+1} va a ser asintóticamente mayor que una constante no nula. Éste es un requisito mínimo, pero no debería sorprendernos la aparición de ciertas condiciones que aseguren un cierto tipo de decaimiento en el infinito de los coeficientes α_t.

Establecido esto, presentamos un resultado estándar en teoría de aproximaciones estocásticas que nos permitirá abordar nuestro problema original (B.1). La idea que subyace en él es extender al caso estocástico los métodos iterativos de gradiente deterministas utilizados para hallar ceros de funciones, y la asunción principal es que el ruido nos lleva en media hacia valores más pequeños de la f.

[B.0.1] Teorema. *Sea la definición iterativa de variables aleatorias dada por*

$$r_{t+1} = r_t + \alpha_t s_t.$$

Consideramos una función $f : \mathbb{R}^n \to \mathbb{R}$ mayor o igual que 0 de clase C^1 con ∇f Lipschitz que satisface la siguiente condición:

$$\exists c > 0 / \forall t \in \mathbb{N}_0 \; c\|\nabla f(r_t)\|_2^2 \leq -\nabla f(r_t) \cdot \mathbb{E}\left[s_t | \mathcal{F}_t\right].$$

Exigiremos que el término aleatorio $\{s_t\}_{t\in\mathbb{N}_0}$ cumpla la condición de acotación

$$\mathbb{E}\left[\|s_t\|^2 \middle| \mathcal{F}_t\right] \leq A + K\|\nabla f(r_t)\|_2^2,$$

donde A es una variable aleatoria finita en casi todo ω y K es una constante mayor que 0.

Por último, impondremos que los $\alpha_t \geq 0$ sean $\mathcal{F}_t$-medibles y cumplan que $\sum_{t=0}^{\infty} \alpha_t = \infty$ y $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ con probabilidad 1. Entonces, con probabilidad 1 se tiene que

- *La sucesión $\{f(r_t)\}_{t\in\mathbb{N}_0}$ converge;*
- *Se cumple que $\lim_{t\to\infty} \nabla f(r_t) = 0$;*
- *Los puntos de acumulación de $\{r_t\}_{t\in\mathbb{N}_0}$ son estacionarios de f.*

La prueba se puede encontrar en [4] para el caso en el que la variable aleatoria A es determinista. No obstante, esa misma demostración es valida para la extensión al caso que nos incumbe aquí.

Ese resultado se puede particularizar para llegar al siguiente corolario, que ahora sí aborda de manera directa las situaciones con las que nos vamos a topar.

[B.0.2] Corolario. *Consideremos la iteración estocástica*

$$r_{t+1} = (1 - \alpha_t) r_t + \alpha_t w_t.$$

Si los valores de α_t cumplen las condiciones del teorema anterior y las perturbaciones $\{w_t\}_{t\in\mathbb{N}_0}$ son tales que

- $\mathbb{E}\left[w_t|\mathcal{F}_t\right] = 0$
- $\mathbb{E}\left[w_t^2|\mathcal{F}_t\right] \leq A + Kr_t^2$ *siendo* A *una v.a. finita en casi todo punto y* $K \geq 0$ *una constante.*

Se tiene entonces que $\lim_{t\to\infty} r_t = 0$ *con probabilidad 1.*

[B.0.2] *Demostración.* Basta escribir la iteración como $r_{t+1} = r_t + \alpha_t(w_t - r_t)$ y ver que se cumplen las condiciones para aplicar el teorema anterior con $f = \frac{|x|^2}{2}$.

□

A modo de ejemplo de uso de estos resultados, veamos su capacidad para dar lugar a métodos de estimación. Éste será justo el fin con el cuál los utilizaremos en este trabajo; resulta natural, pues, verlos primero en acción para uno de los ejemplos de estimación más sencillo, el del valor de la media.

[B.0.3] Ejemplo. *Sean* $\{v_t\}_{t\in\mathbb{N}_0}$ *una familia de variables aleatorias independientes e idénticamente distribuidas con media* μ *y varianza* $\sigma^2 < \infty$. *Si escogemos una sucesión* $\{\alpha_t\}_{t\in\mathbb{N}_0}$ *que satisfaga las condiciones del teorema anterior, se cumplirá que la iteración*

$$r_{t+1} = (1 - \alpha_t)r_t + \alpha_t v_t \tag{B.2}$$

convergerá a $\lim_{t\to\infty} r_t = \mu$.

Para verlo, basta restar μ *a ambos lados de (B.2) para llegar a*

$$\tilde{r}_{t+1} = (1 - \alpha_t)\tilde{r}_t + \alpha_t\tilde{v}_t,$$

donde $\{\tilde{r}_t\}_{t\in\mathbb{N}_0} = \{r_t - \mu\}_{t\in\mathbb{N}_0}$ *y* $\{\tilde{v}_t\}_{t\in\mathbb{N}_0} = \{v_t - \mu\}_{t\in\mathbb{N}_0}$. *Las variables aleatorias* $\{\tilde{v}_t\}_{t\in\mathbb{N}_0}$ *siguen siendo independientes idénticamente distribuidas y con varianza finita, pero ahora su valor esperado es* 0. *Por tanto, el Corolario B.0.2 aplica y nos permite concluir que* $\lim_{t\to\infty} \tilde{r}_t = \lim_{t\to\infty} r_t - \mu = 0$.

Bibliografía

[1] Mohammad Gheshlaghi Azar, Remi Munos, M Ghavamzadaeh, and Hilbert J Kappen. Speedy q-learning. 2011.

[2] Martino Bardi and Italo Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Springer Science & Business Media, 2008.

[3] Richard Bellman. *Dynamic programming*. Princeton University Press, 1957.

[4] Dmitri P. Bertsekas and John N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, Belmont, MA, 1996.

[5] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[6] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[7] Piermarco Cannarsa and Carlo Sinestrari. *Semiconcave functions, Hamilton-Jacobi equations, and optimal control*, volume 58. Springer Science & Business Media, 2004.

[8] Kai Lai Chung. Markov chains. *Springer-Verlag, New York*, 1967.

[9] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[10] Michael G Crandall and Pierre-Louis Lions. Viscosity solutions of hamilton-jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

[11] Cyrus Derman. *Finite state Markovian decision processes*. Academic Press, Inc., 1970.

[12] Cyrus Derman and Ralph E Strauch. A note on memoryless rules for controlling sequential control processes. *The Annals of Mathematical Statistics*, pages 276–278, 1966.

[13] Adithya M Devraj and Sean P Meyn. Zap q-learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2232–2241, 2017.

[14] Stuart Dreyfus. Richard bellman on the birth of dynamic programming. *Operations Research*, 50(1):48–51, 2002.

[15] Carlos Esteve Yague. An introduction to Reinforcement Learning and Optimal Control Theory. https://cmc.deusto.eus/an-introduction-to-reinforcement-learning-and-optimal-control-theory/, 2020. [Online; accedido el 01-Junio-2021].

[16] Carlos Esteve Yague. Q-Learning for finite-dimensional problems. https://deustotech.github.io/DyCon-Blog/tutorial/dp00/P0004, 2020. [Online; accedido el 01-Junio-2021].

[17] Lawrence C Evans. Partial differential equations. *Graduate studies in mathematics*, 19(2), 1998.

[18] Eyal Even-Dar and Yishay Mansour. Convergence of optimistic and incremental q-learning. In *NIPS*, pages 1499–1506, 2001.

[19] Eyal Even-Dar, Yishay Mansour, and Peter Bartlett. Learning rates for q-learning. *Journal of machine learning Research*, 5(1), 2003.

[20] Wendell H Fleming and Halil Mete Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

[21] Gerald B Folland. *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons, 1999.

[22] Iosif Il'ich Gihman and Anatolij Vladimirovič Skorohod. *Controlled stochastic processes*. Springer Science & Business Media, 2012.

[23] Skorokhod Anatoli V. Gikhman, Iosif I. *The theory of stochastic processes I*. Springer-Verlag, 1974.

[24] Abhijit Gosavi. Boundedness of iterates in q-learning. *Systems & control letters*, 55(4):347–349, 2006.

[25] Onésimo Hernández-Lerma and Jean B Lasserre. *Discrete-time Markov control processes: basic optimality criteria*, volume 30. Springer Science & Business Media, 2012.

[26] Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.

[27] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[28] Michael Kearns and Satinder Singh. Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in neural information processing systems*, pages 996–1002, 1999.

[29] Francisco S Melo. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pages 1–4, 2001.

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[31] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[32] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[33] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[34] James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.

[35] Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 881–888, 2006.

[36] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[37] Csaba Szepesvári et al. The asymptotic convergence-rate of q-learning. In *NIPS*, volume 10, pages 1064–1070. Citeseer, 1997.

[38] William F Trench. Conditional convergence of infinite products. *The American mathematical monthly*, 106(7):646–651, 1999.

[39] John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16(3):185–202, 1994.

[40] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[41] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.