UNIVERSITAT DE BARCELONA
Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

Optimización con programación dinámica

Autor: Keila Ruth Rosell Esau

Director: Dr. José Manuel Corcuera Valverde
Realizado a: Departamento de Matemáticas e Informática
Barcelona, 23 de enero de 2022

Abstract

In this thesis we study the optimization method called Dynamic Programming and how it is implemented to solve sequential problems, that is, those problems in which the solution is to make a series of decisions in many different stages in order to maximize a reward, according to a purpose. Different approaches are analyzed, depending on whether all the data is known for the problem, in the deterministic case, or if the data is determined by a probability distribution, in the stochastic case. A distinction will also be made for cases where time evolves in a discrete way or if it does so continuously. For each case we will develop the Hamilton-Jacobi-Bellman equation, which is a central element of the dynamic programming algorithms and is useful in finding and comparing different strategies for the decision-making agent. Finally, dynamic programming is applied to reinforcement learning, which is an area of artificial intelligence that is focused on determining what actions a software agent must choose in a given environment, in order to find the highest reward.

Resumen

En este trabajo se estudia el método de optimización llamado Programación Dinámica y cómo se implementa para resolver problemas de tipo secuencial, es decir, aquellos problemas en los que la solución es tomar una serie de decisiones en diferentes etapas con tal de maximizar alguna noción de recompensa, de acuerdo a un propósito. Se analizan diferentes enfoques, dependiendo si en el problema se conocen todos los datos, en el caso determinista, o si quedan determinados mediante una distribución de probabilidad, en el caso estocástico. También se hará distinción en los casos donde el tiempo evoluciona de forma discreta o si lo hace de forma continua. Para cada caso se desarrolla la ecuación de Hamilton-Jacobi-Bellman, que es un elemento central de los algoritmos de la programación dinámica y que sirve para encontrar y comparar diferentes estrategias para el agente que toma las decisiones. Finalmente se aplica la programación dinámica al aprendizaje por refuerzo, que es una área de la inteligencia artificial que está centrada en determinar qué acciones debe escoger un agente de software en un entorno dado con el fin de encontrar la máxima recompensa.

2020 Mathematics Subject Classification. 49L12, 49L20, 60J05

Agradecimientos

Me gustaría empezar dando las gracias a mi tutor, el Dr. José Manuel Corcuera Valverde, por guiarme a lo largo de este trabajo y por su disponibilidad.

En segundo lugar, quiero agradecer a mi familia y amigos, por acompañarme a lo largo de estos meses y por la confianza que han tenido siempre en mi.

Por último, también quiero dar gracias a Dios. Es gracias a Él que he podido llegar hasta aquí y es quien le da sentido a todo lo que estudio.

Índice

1. Introducción

Uno de los problemas que más se ha planteado es cómo un agente puede aprender a predecir y controlar la respuesta de su entorno. Esta disciplina se conoce comúnmente como *control óptimo*. Cuando el modelo del entorno es conocido, se dice que entonces el agente puede planificar su secuencia de decisiones óptimas. Estos son problemas de toma de decisiones en los que los costos y beneficios se distribuyen a lo largo del tiempo y donde las decisiones tomadas en un momento determinado influencian las posibilidades disponibles en otros puntos del tiempo. Este tipo de problemas de elección intertemporal se ve en muchas aplicaciones como por ejemplo en el campo de las finanzas, en problemas como la fijación de precios, inversión y asignación de carteras, la creación de mercado, etc.

En este trabajo queremos desarrollar el vínculo entre estos problemas de optimización dinámica y el aprendizaje por refuerzo (*reinforcement learning*, RL), modelos de aprendizaje automático que permiten a los agentes de RL aprender a tomar una secuencia de decisiones a través de "ensayo y error", incorporando el *feedback* de sus acciones y experiencias.

Algo que es clave para estos problemas de optimización dinámica es determinar las mejores acciones posibles que maximicen el valor entre dos o más recompensas en diferentes instantes de tiempo. El enfoque más común para resolver problemas de optimización dinámica es la programación dinámica (*dynamic programming*, DP). DP se refiere a una colección de algoritmos que se pueden utilizar para encontrar explícitamente soluciones a la ecuación de Bellman, una ecuación que desarrollaremos más adelante y que es fundamental en la programación dinámica. Con esta ecuación se podrán encontrar estrategias óptimas, es decir secuencias de decisiones que nos llevaran a tomar las mejores acciones y recibir el máximo beneficio posible.

Mientras que DP puede ser aplicado en un entorno determinista o estocástico, y en tiempo discreto o continuo, se basa en varias suposiciones que rara vez son ciertas en la práctica: (i) Se conoce con precisión la dinámica del entorno, (ii) se tienen suficientes recursos computacionales para realizar el cálculo de la solución y (iii) se cumple la propiedad de Markov, que veremos más adelante. En el mundo real, para muchas de las aplicaciones generalmente no se podrá implementar la solución DP exactamente porque una o varias de estas suposiciones no se cumplirán. De hecho, estas suposiciones se incumplen incluso en muchos juegos simples. Por ejemplo, un juego de mesa de dos jugadores podría tener tantos posibles estados que haría muy difícil resolver la ecuación de Bellman y se incumpliría (ii). O en muchos problemas no se dispone de un modelo completo del sistema y por tanto se incumpliría (i), por ejemplo en el área de las finanzas donde muchos sistemas estocásticos son complejos y es difícil derivar o estimar expresiones correctas de su dinámica.

El aprendizaje por refuerzo proporciona una forma de superar estos problemas por medio de la construcción de agentes que actúan de forma inteligente, dando lugar a soluciones eficientes a problemas que no se pueden resolver usando únicamente DP.

La primera parte del trabajo estará enfocada en introducir la programación dinámica y los diferentes elementos necesarios para después poder desarrollar las ecuaciones que

nos permitirán resolver los problemas. La segunda parte está enfocada en la aplicación de la programación dinámica en el aprendizaje por refuerzo (RL), dando varios algoritmos y ejemplos. También veremos cuáles son algunas de las limitaciones de RL.

2. Introducción a la programación dinámica

La Programación Dinámica (*Dynamic Programming*, DP) es un método de optimización, es decir, un método usando ecuaciones matemáticas y algoritmos para resolver problemas con el propósito de encontrar la mejor solución posible entre todas las soluciones viables. Este método fue inventado en 1953 por el matemático Richard Bellman. La DP inicialmente se desarrolló para la resolución de problemas en procesos de decisión en múltiples pasos, pero las mismas ideas pueden utilizarse en otros tipos de problemas de matemática aplicada o para el planteo de algunas cuestiones teóricas.

La idea de Bellman sobre la teoría de programación dinámica se basa en una estructura de optimización, la cual consiste en descomponer el problema en subproblemas con resolución más asequible. Los cálculos se realizan recursivamente, usando la solución óptima de un subproblema como dato de entrada del siguiente. Por lo cual, se entiende que el problema es solucionado en su totalidad una vez se haya solucionado el último subproblema. Dentro de esta teoría, Bellman desarrolla el *Principio de Optimalidad*, del que más tarde hablaremos, el cual es fundamental para la resolución adecuada de los cálculos recursivos. Es por ello, que se define a la programación dinámica como una técnica matemática que ayuda a resolver decisiones secuenciales interrelacionadas, combinándolas para obtener la solución óptima.

Existen dos enfoques en la Programación Dinámica:

- Programación Dinámica Determinista: Consiste en que el estado de la siguiente etapa se encuentra determinado por completo con respecto al estado y la decisión que posee la etapa actual.
- Programación Dinámica Estocástica: En este enfoque, el estado de la siguiente etapa y estrategia de decisión queda completamente determinado mediante una distribución de probabilidad.

2.1. Historia de la programación dinámica

Richard Ernest Bellman fue un matemático aplicado, cuya mayor contribución fue la metodología denominada programación dinámica, inventada en 1953. Nació el 26 de agosto de 1920 en Brooklyn (Estados Unidos) y murió el 19 de marzo de 1984 en Los Ángeles.

Bellman completó sus estudios en la *Abraham Lincoln High School* en 1937, y estudió matemáticas en el *Brooklyn College*, donde obtuvo su licenciatura en 1941. Más tarde obtuvo una maestría de la *Universidad de Wisconsin-Madison*. Durante la Segunda Guerra Mundial trabajó para un grupo de Física Teórica de la División de Los Álamos. En 1946 recibió su doctorado en Filosofía (Ph.D.) de *Princeton* bajo la supervisión de Salomón Lefschetz. A partir de 1949 Bellman trabajó durante muchos años en la corporación RAND y fue durante ese tiempo cuando desarrolló la programación dinámica.

Fue profesor en la Universidad del Sur de California, miembro de la Academia Americana de las Artes y las Ciencias (1975), y miembro de la Academia Nacional de Ingeniería (1977). Además, fue galardonado con la Medalla de Honor del IEEE en 1979, ≪por sus

contribuciones a los procesos de decisión y la teoría de sistemas de control, en particular por la creación y aplicación de la programación dinámica≫. Su obra fundamental es la ecuación de Bellman.

La ecuación de Bellman, también conocida como la ecuación de la programación dinámica, es una condición necesaria para la optimalidad asociada con el método de optimización matemática conocido como la programación dinámica. Casi cualquier problema que se pueda resolver utilizando la teoría de control óptimo también se puede resolver mediante el análisis de la correspondiente ecuación de Bellman. Esta ecuación se aplicó por primera vez a la teoría de la ingeniería de control y otros campos de la matemática aplicada, y posteriormente se convirtió en una herramienta importante en la teoría económica.

Otro resultado importante en la programación dinámica y que también fue desarrollado por Richard Bellman, con compañeros de trabajo, fue la ecuación de Hamilton-Jacobi-Bellman (HJB). Esta ecuación es una ecuación diferencial parcial que es fundamental para la teoría de control óptimo.

A lo largo de su carrera, Bellman publicó 619 artículos y 39 libros. Uno de sus libros más conocidos es precisamente el de 1957, *Programación Dinámica*. En este libro clásico, Bellman introduce al lector a la teoría matemática de su tema, la programación dinámica. Su objetivo es mostrar cómo los procesos de decisión de múltiples etapas, que ocurren en varios tipos de situaciones de interés (como en el campo comercial, industrial, económico, etc) son susceptibles de análisis matemático. Fue escrito en la época de las computadoras digitales de alta velocidad y gran capacidad. Es por eso que en este libro toma como objetivo la deducción matemática de la estructura de estrategias de decisión óptimas para tales problemas.

2.2. Definición del problema y notación

A continuación introduciremos algunos conceptos básicos que usaremos a lo largo de todo el trabajo y que se usaran para plantear diferentes tipos de problemas en este campo, y la notación que los acompaña.

Cuando un agente se enfrenta a un problema, lo hace mediante la toma de decisiones, escogiendo acciones. Su objetivo es elegir una secuencia de acciones que haga que el sistema funcione de manera óptima con respecto a algún criterio de rendimiento predeterminado. Las decisiones que se toman deben estar bien analizadas ya que hay que anticipar las oportunidades y los costes (o recompensas) asociados a los estados de futuras etapas.

2.2.1. Estados y acciones

Antes de explicar los conceptos de *estado* y *acción* es necesario definir los instantes de decisión.

Definición 2.1. *Los **instantes de decisión** son los puntos en el tiempo en los cuales se toman decisiones. Denotamos a este conjunto como $\mathcal{T}$.*

Podemos clasificar este conjunto $\mathcal{T}$ de dos formas: como un conjunto discreto o uno

continuo. Cuando nos encontramos en un conjunto discreto, el agente toma decisiones en cada uno de los instantes $t \in \mathcal{T}$. En cambio, cuando es un conjunto continuo, las decisiones se toman en puntos aleatorios del tiempo cuando ocurren determinados eventos o en instantes de tiempo elegidos por el agente. Con esto podemos formular modelos en los que un instante de decisión corresponde al inicio de una etapa, y cada etapa viene definida por una serie de elementos (estado, acción, recompensa...) que a continuación definiremos. En el caso discreto, el conjunto de decisiones puede ser finito ($\mathcal{T} = \{1, 2, ..., T\}$ con $T < \infty$) o infinito ($\mathcal{T} = \mathbb{N}$). En el caso continuo $\mathcal{T}$ es un intervalo $\mathcal{T} = [0, T]$ o $\mathcal{T} = [0, \infty)$.

Definimos los conceptos de estado y acción.

- **Estado**: Es la representación de una etapa en un momento determinado, es decir, de qué información disponemos y en qué situación nos encontramos. Usaremos S para denotar el estado general, $\mathcal{S}$ para denotar conjuntos de valores que puede tomar S y $s \in \mathcal{S}$ para una estado específico. A veces usaremos s' para referirnos al siguiente estado y s_t para referirnos a un estado en un tiempo concreto.
- **Acción**: En un instante de decisión concreto, el agente observa el sistema en un estado $s \in \mathcal{S}$ y escoge una acción a del conjunto de acciones permisibles en el estado s, que denotamos $\mathcal{A}_s$. Usamos A para referirnos al conjunto general de acciones, entonces A toma valores en $\bigcup_{s \in \mathcal{S}} \mathcal{A}_s$. Igual que antes, usaremos a' para referirnos a la siguiente acción y cuando queremos enfatizar en qué tiempo es tomada la acción, escribiremos a_t. Observamos que ejecutar una cierta acción no siempre garantiza el estado en el que acabaremos.

Los conjuntos $\mathcal{S}$ y $\mathcal{A}_s$ pueden ser:

- Conjuntos finitos arbitrarios.
- Conjuntos contables infinitos arbitrarios.
- Subconjuntos compactos de espacios euclidianos de dimensión finita.
- Subconjuntos no vacíos de Borel, es decir, subconjuntos obtenidos mediante uniones e intersecciones numerables, de espacios métricos completos y separables.

2.2.2. Recompensas

Cuando el agente realiza una acción puede recibir una recompensa (beneficio a su favor) o una penalización. Esta recompensa es un resultado de elegir la acción $a \in \mathcal{A}$, en el estado $s \in \mathcal{S}$, en el instante de decisión $t \in \mathcal{T}$. Por tanto, escribiremos $R := R_t(\mathcal{S}, \mathcal{A})$.

Si el valor de R es positivo, lo tomamos como una recompensa que nos beneficia y cuando es negativo, como un coste, es decir, una recompensa que nos perjudica. Cada proceso es diferente, pero observamos que las recompensas normalmente se reciben al finalizar cada etapa, es decir después de cada decisión. Solo se requiere que se conozca su valor o su valor esperado antes de elegir una acción, y que no se vea afectado por acciones futuras. La recompensa puede ser:

- Una suma global recibida en un momento fijo o aleatorio antes del siguiente instante de decisión.
- Acumulado de forma continua a lo largo del periodo actual.
- Una cantidad aleatoria que depende del estado del sistema en el instante de decisión posterior.
- Una combinación de las anteriores.

También podemos escribir,

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

Es decir, la recompensa esperada sabiendo el estado actual y el siguiente, y la acción que realizamos.

Definimos el *retorno* como el total de las recompensas que nos esperan. El objetivo del agente es maximizar la recompensa total, es decir, el retorno G_t, que en el caso más simple es $G_t = R_{t+1} + R_{t+2} + ... + R_T$ cuando $\mathcal{T}$ es finito. Pero si no tenemos punto terminal, la fórmula de arriba presenta problemas, ya que no podemos hacer el cálculo cuando $T = \infty$. Podemos usar entonces la forma alternativa donde se van haciendo cada vez más pequeñas las contribuciones de las recompensas más lejanas:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

En esta fórmula el parámetro $\gamma \in [0, 1]$ se conoce como *razón de descuento* y se usa para representar que cuanto antes recibimos la recompensa, más valor tiene.

Además, existen diferentes tipos de modelos, dependiendo de lo que buscamos optimizar, suponiendo de momento que $\mathcal{T}$ es discreto:

- Modelo de horizonte finito: el agente trata de optimizar su recompensa esperada en los siguientes T instantes de decisión, sin preocuparse de lo que pueda ocurrir después:
$$\mathbb{E}\left(\sum_{t=0}^{T} R_t\right)$$
- Modelo de horizonte infinito: las recompensas que recibe el agente son reducidas geométricamente de acuerdo al factor de descuento $\gamma \in [0, 1)$, pudiendo considerar así un número infinito de pasos:
$$\mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R_t\right)$$
- Modelo con recompensa promedio: el agente optimiza a largo plazo la recompensa promedio:
$$\lim_{T\to\infty} \mathbb{E}\left(\frac{1}{T}\sum_{t=0}^{T} R_t\right)$$

Por último, definimos dos conceptos más que tienen mucha relación con el concepto de recompensa.

- **Función valor**: Es una función que mide cómo de buena o mala es una acción o un estado. Si nos encontramos en un cierto estado y todos los estados futuros son buenos, entonces el valor en nuestro estado actual será alto. Este valor depende de qué acciones tomamos en el presente y en el futuro. El *valor* viene dado por la acumulación de recompensas y presenta cómo de buena es nuestra posición actual dada la acción que tomamos ahora y en el futuro. Más adelante definiremos formalmente dos tipos de función valor: función valor del estado y función valor de la acción.
- **Probabilidad de transición**: Es la probabilidad que determina el estado del sistema en el instante de decisión siguiente al actual, es decir, la probabilidad de ir de un estado s al estado s' después de tomar una acción a. Por tanto, podemos escribir esta probabilidad como:

$$\mathcal{P}_{ss'}^{a} := \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$$

2.2.3. Estrategias

Una *regla de decisión* es un procedimiento que sirve para seleccionar una acción en cada estado en un instante de decisión específico. Es una función $d_t : \mathcal{S} \longrightarrow \mathcal{A}$ que especifica qué acción tomar cuando el sistema ocupa un estado s en el instante de decisión t. Para cada estado $s \in \mathcal{S}$ tenemos una norma de decisión $d_t(s) \in \mathcal{A}_s$. Denotamos al conjunto de reglas de decisión en un tiempo t como D_t. Estas reglas de decisión pueden ser Markovianas o dependientes de la historia del proceso, y deterministas o aleatorias.

- Markovianas: Cuando la regla de decisión depende de los estados y acciones anteriores, pero solo a través del actual estado del sistema, es decir, sin importar cómo se ha llegado al estado actual.
  - Deterministas (D_t^{MD}): Cuando la decisión se basa en elegir una acción con certeza, es decir, $d_t(s_t) \in \mathcal{A}_{s_t}$.
  - Aleatorias (D_t^{MA}): Cuando la decisión se basa en especificar una distribución de probabilidad $p(.|s_t)$ en el conjunto de acciones. Por tanto, $p(.|s_t) \in \mathcal{M}_{\mathbb{P}}(\mathcal{A}_{s_t})$, donde $\mathcal{M}_{\mathbb{P}}$ denota las medidas de probabilidad, en este caso en $\mathcal{A}_{s_t}$.
- No Markovianas: Cuando la regla de decisión depende de toda la historia del proceso, es decir, de la secuencia de estados y acciones previas. En este caso, la función d_t es una función del historial $h_t = (s_1, a_1, ..., s_{t-1}, a_{t-1}, s_t)$, donde s_i y a_i denotan el estado y la acción del sistema en el instante de decisión i.
  - Deterministas (D_t^{HD}): Cuando la decisión se basa en elegir una acción con certeza. En este caso $d_t(h_t) \in \mathcal{A}_{s_t}$.
  - Aleatorias (D_t^{HA}): Cuando la decisión se basa en especificar una distribución de probabilidad $p(.|h_t)$ en el conjunto de acciones $\mathcal{A}_{s_t}$. En este caso, $p(.|h_t) \in \mathcal{M}_{\mathbb{P}}(\mathcal{A}_{s_t})$.

A partir de ahora nos centraremos en las Markovianas, tanto deterministas como aleatorias.

Una *estrategia* especifica qué regla de decisión debemos tomar en cada instante de decisión, es decir, provee al agente de un procedimiento para seleccionar una acción bajo cualquier estado futuro. Formalmente:

Definición 2.2. *Una **estrategia** π es una secuencia de reglas de decisión $\pi = (d_1, d_2, ..., d_T)$ donde $d_t \in D_t$ para $t = 1, 2, ..., T$.*

Las estrategias, igual que las reglas de decisión, pueden ser deterministas o aleatorias, y Markovianas o no Markovianas. Las estrategias también pueden ser estacionarias o no-estacionarias. Una estrategia estacionaria decide la misma acción para cada estado independientemente del tiempo, esto es $a_t(s) = a(s)$ para toda $t \in \mathcal{T}$; lo cual no se cumple para una no-estacionaria.

Con todas estas definiciones, podemos establecer ahora lo que es el *Proceso de decisión Markoviano* que veremos a continuación.

2.3. Proceso de decisión Markoviano

Cuando un agente se enfrenta a un problema de toma de decisiones bajo incertidumbre y con información limitada del ambiente, éste debe tomar la mejor decisión de acuerdo a sus objetivos. Muchas veces este proceso se repite de forma secuencial en el tiempo, de forma que en cada instante t el agente recibe información y decide qué acción tomar. A estos problemas se los denomina *problemas de decisión secuencial*. Aquí es donde entran los *Procesos de Decisión Markovianos*, que son modelos para la toma de decisiones en este tipo de problemas, es decir, cuando se dispone de información limitada.

Definición 2.3. *Un **Proceso de Decisión Markoviano** (Markov decision process, MDP) es un conjunto $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P} >$, donde cada uno de los elementos son:*

- *Un conjunto finito de estados $\mathcal{S} = \{s_1, s_2, ..., s_n\}$, donde s_t denota el estado $s \in \mathcal{S}$ en el instante t.*
- *Un conjunto de acciones $\mathcal{A}$, que consideramos también finito. Para cada $s \in \mathcal{S}$, tenemos $\mathcal{A}(s) \subseteq \mathcal{A}$, con $\mathcal{A}(s) = \{a_1(s), a_2(s), ..., a_n(s)\}$, donde $a_t(s)$ denota la acción realizada en un estado s en el instante t.*
- *Una función de recompensa $\mathcal{R}$ que devuelve la recompensa recibida en cada estado $s' \in \mathcal{S}$ dado que en el estado anterior $s \in \mathcal{S}$ se realiza la acción $a \in \mathcal{A}(s)$.*
- *Una probabilidad de transición $\mathcal{P}$ que devuelve la probabilidad de llegar a cada estado $s' \in \mathcal{S}$ dado que se tomó la acción $a \in \mathcal{A}(s)$ en $s \in \mathcal{S}$.*

Las acciones del agente determinan no sólo la recompensa inmediata, sino también la probabilidad del siguiente estado. En los MDP, los procesos son *markovianos* tal y como su nombre indica; esto significa que el siguiente estado es independiente de los estados anteriores una vez se ha fijado el estado actual y la acción que se toma. Por tanto, no tiene importancia cómo hemos llegado al estado en el que nos encontramos, sino solo la información que disponemos del estado actual y la acción que tomamos. Podemos escribir esto formalmente como:

$$\mathcal{P}^a_{ss'} = \mathbb{P}(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, ...)$$

$$= \mathbb{P}(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

El problema fundamental en un MDP es encontrar una estrategia óptima π^*, es decir, aquella que maximiza la recompensa que espera recibir el agente a largo plazo.

3. Ecuaciones de Bellman

En el capítulo anterior hemos visto que un agente necesita una estrategia para saber qué regla de decisión tomar y realizar la mejor acción. En este capítulo presentaremos las funciones valor y desarrollaremos las ecuaciones de Bellman, que son elementos fundamentales para los algoritmos de la programación dinámica para encontrar y comparar estrategias para los agentes. El objetivo principal es encontrar una estrategia óptima para cada estado.

Cuando tratamos con MDPs, en programación dinámica se consideran dos tipos de problemas diferentes:

- Predicción: consiste en encontrar la función valor del estado o de la acción, que a continuación detallaremos, para una estrategia dada. Este problema se conoce típicamente como evaluación de la estrategia (*policy evaluation*) ya que la función valor nos dirá cómo de buena es una estrategia (en términos de la recompensa acumulada esperada).
- Control: consiste en encontrar la estrategia óptima, es decir, la estrategia que lleva a la función valor óptima.

Antes de definir las funciones valor, es importante exponer el Principio de Optimalidad de Bellman.

3.1. Principio de Optimalidad de Bellman

Suponemos que el agente se enfrenta a un problema de toma de decisiones. Como vemos en la Figura 1, el agente se encuentra inicialmente en un estado s_1. Toma la primera decisión, es decir la acción $a(s_1)$, que resulta en el segmento $s_1 - s_2$ con una recompensa de $\mathcal{R}_{s_1 s_2}$. Las decisiones restantes dan lugar al segmento $s_2 - s_n$, con recompensa $\mathcal{R}_{s_2 s_n}$. La recompensa máxima para ir de s_1 a s_n es

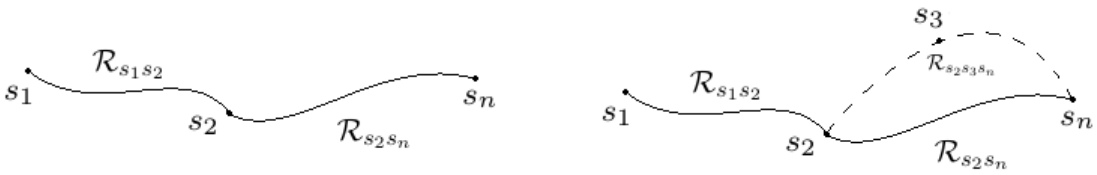
$$\mathcal{R}^*_{s_1 s_n} = \mathcal{R}_{s_1 s_2} \times \mathcal{R}_{s_2 s_n}$$

donde "$\times$" indica la operación mediante la cual se acumulan las recompensas, es decir, las recompensas se pueden acumular de forma aditiva, multiplicativa, etc.

Afirmación: Si $s_1 - s_2 - s_n$ es el camino óptimo de s_1 a s_n, es decir si son los estados por los que debemos pasar para ir de s_1 a s_n para obtener la recompensa máxima, entonces $s_2 - s_n$ es el camino óptimo de s_2 a s_n.

Demostración. Lo demostramos por contradicción. Suponemos que $s_2 - s_3 - s_n$ es el camino óptimo de s_2 a s_n. Entonces la recompensa de este será mayor que cualquier otro

Figura 1: Camino óptimo



camino entre estos dos estados

$$\mathcal{R}_{s_2 s_3 s_n} > \mathcal{R}_{s_2 s_n} \Rightarrow \mathcal{R}_{s_1 s_2} \times \mathcal{R}_{s_2 s_3 s_n} > \mathcal{R}_{s_1 s_2} \times \mathcal{R}_{s_2 s_n} = \mathcal{R}^*_{s_1 s_n}$$

Pero esto contradice la condición de que $s_1 - s_2 - s_n$ es el camino óptimo de s_1 a s_n. □

Esta propiedad que acabamos de demostrar es la que Bellman usa para exponer su principio de optimalidad, que es el siguiente:

Principio de Optimalidad de Bellman. Una estrategia óptima tiene la propiedad de que cualquiera sean el estado y la decisión iniciales, las decisiones siguientes constituyen una estrategia óptima con respecto al estado de la primera decisión. Es decir, buscamos un óptimo fijadas las condiciones anteriores.

Observamos que para procesos que no son Markovianos, es decir aquellos en los que las recompensas y la dinámica dependen no solo del estado actual, sino de la historia previa, las técnicas de solución desarrolladas para MDP no se pueden aplicar directamente y el problema es más complejo. Para estos casos, se pueden usar variables temporales para especificar la dependencia de la historia, y, agregando las variables adecuadas, se puede convertir en un MDP equivalente.

3.2. Funciones valor

Para encontrar y comparar estrategias, el agente intenta evaluar el "valor"de los estados y de las acciones que puede realizar en relación al objetivo que desee alcanzar. Para ello, nos basamos en una función valor que como ya comentamos anteriormente, es una función que determina lo que es "bueno"para el agente a largo plazo.

Dado que el agente tiene por objetivo encontrar la estrategia que maximiza la recompensa a largo plazo, es decir, la función valor, el proceso de decisión debe estar guiado por esta función más que por las recompensas instantáneas R_t. Nos interesa estimar la función valor ya que, una vez conocida, podremos mejorar la estrategia disponible.

Hay dos tipos de funciones valor: la *función valor del estado* y la *función valor de la acción*.

- Función valor del estado (*state-value function* o *V-function*): Esta función mide la bondad de cada estado. Nos dice el retorno G_t con descuento γ que podemos esperar en el futuro partiendo de un estado $s \in \mathcal{S}$. En otras palabras, nos dice lo bueno o

malo que es estar en un estado particular de acuerdo con el retorno con descuento cuando seguimos una determinada estrategia.

- Función valor de la acción (*action-value function* o *Q-function*): Esta función se obtiene al ampliar la definición de función valor del estado definiendo un valor para cada acción $a \in \mathcal{A}_s$ que se puede realizar desde un estado s. Esta función indica lo bueno o malo que es realizar una acción específica del conjunto de acciones que podemos elegir desde el estado en el que nos encontramos.

Función valor del estado

En términos generales, podemos decir que esta función responde a la pregunta básica de "¿Qué esperar si estamos aquí?". Formalmente, la función valor del estado mide la bondad de cada estado $s \in \mathcal{S}$ según el retorno con descuento G_t al seguir una estrategia π determinada. Entonces, podemos definir esta función como la recompensa total esperada (descontada o no descontada, según el valor γ) que se puede obtener del estado siguiendo una estrategia π. Tenemos entonces la función $V_\pi : \mathcal{S} \longrightarrow \mathbb{R}$ donde

$$V_\pi(s_t) = \mathbb{E}_\pi \left(G_t | S_t = s_t\right) = \mathbb{E}_\pi \left(\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s_t \right)$$

Función valor de la acción

Por otro lado, como hemos mencionado antes, la función valor de la acción define el valor de realizar una acción determinada $a \in \mathcal{A}_s$ en un estado $s \in \mathcal{S}$ concreto de acuerdo con la estrategia π que estamos siguiendo. Podemos expresar esta función como $Q_\pi : \mathcal{S} \times \mathcal{A}_s \longrightarrow \mathbb{R}$ donde

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi \left(G_t | S_t = s_t, A_t = a_t\right) = \mathbb{E}_\pi \left(\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s_t, A_t = a_t \right)$$

Esta expresión describe el retorno con descuento esperado al realizar una acción a, comenzando desde el estado s en el instante t y siguiendo la estrategia π.

3.3. Desarrollo de las ecuaciones de Bellman

A partir de las expresiones de las funciones valor $V_\pi(s_t)$ y $Q_\pi(s_t, a_t)$ descritas anteriormente, podemos escribirlas de forma recursiva como:

$$\begin{aligned} V_\pi(s_t) &= \mathbb{E}_\pi[G_t | S_t = s_t] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s_t] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + ...) | S_t = s_t] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s_t] \end{aligned}$$

Vemos entonces que el valor de un estado s se puede obtener como la suma de la recompensa inmediata y el retorno descontado del siguiente estado.

Y, por la propiedad iterativa de la esperanza condicionada y por ser un proceso markoviano, vemos que

$$\mathbb{E}_\pi[G_{t+1}|S_t = s_t] = \mathbb{E}_\pi\left[\mathbb{E}(G_{t+1}|S_{t+1})|S_t = s_t\right] = \mathbb{E}_\pi\left[V_\pi(S_{t+1})|S_t = s_t\right]$$

Con esto, encontramos la siguiente relación de recursividad:

$$V_\pi(s_t) = \mathbb{E}_\pi[R_{t+1} + \gamma V_\pi(S_{t+1})|S_t = s_t]$$

De la misma manera,

$$\begin{aligned} Q_\pi(s_t, a_t) &= \mathbb{E}_\pi[G_t|S_t = s, A_t = a_t] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...|S_t = s_t, A_t = a_t] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + ...)|S_t = s_t, A_t = a_t] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s_t, A_t = a_t] \end{aligned}$$

Vemos aquí como también se puede descomponer la función Q en la recompensa instantánea más el retorno descontado del par estado-acción sucesor. Y, como hemos hecho antes con la función valor del estado, aplicando aquí también la propiedad iterativa de la esperanza condicionada y la markovianidad, podemos escribir la siguiente relación de recursividad:

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi[R_{t+1} + \gamma Q_\pi(S_{t+1}, A_{t+1})|S_t = s_t, A_t = a_t]$$

Además, podemos encontrar la siguiente relación entre la función valor del estado y la de acción:

$$V_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) Q_\pi(s, a) \tag{3.1}$$

Y encontramos también la siguiente expresión:

$$Q_\pi(s, a) = \sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}(\mathcal{R}^a_{ss'} + \gamma V_\pi(s')) \tag{3.2}$$

Para simplificar la notación, en ambas expresiones usamos que $a = a_t$, $s = s_t$ y $s' = s_{t+1}$.

Demostración. Obtenemos (3.1) considerando que la acción actual se escoge de acuerdo a la estrategia π. Y obtenemos (3.2) a partir de la expresión a la que hemos llegado con la recursividad de Q_π.

Hemos visto por recursividad que $Q_\pi(s_t, a_t) = \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1}|S_t = s_t, A_t = a_t]$ y esto es equivalente a $Q_\pi(s_t, a_t) = \mathbb{E}_\pi[R_{t+1}|S_t = s_t, A_t = a_t] + \gamma\mathbb{E}_\pi[G_{t+1}|S_t = s_t, A_t = a_t]$. Haciendo el cambio de notación $a = a_t$, $s = s_t$ y $s' = s_{t+1}$, por definición de $\mathcal{R}^a_{ss'}$,

$$\mathbb{E}_\pi[R_{t+1}|S_t = s, A_t = a] = \sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}\mathcal{R}^a_{ss'}$$

y por la siguiente relación,

$$\mathbb{E}_\pi[G_{t+1}|S_t = s, A_t = a] = \sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}V_\pi(s')$$

al final obtenemos (3.2). Recordamos que $\mathcal{P}^a_{ss'}$ indica la probabilidad de alcanzar el estado s' al realizar una acción a en el estado s.
□

Finalmente, sustituyendo la expresión de la función Q encontrada en (3.2) en la expresión de la función V en (3.1), llegamos a:

$$V_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \left(\sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'}(\mathcal{R}^a_{ss'} + \gamma V_\pi(s')) \right) \tag{3.3}$$

Esta ecuación es la que llamamos **ecuación de Bellman para la función V**, que relaciona la función valor actual con la función valor de la siguiente etapa. Y para encontrar la **ecuación de Bellman para la función Q**, sustituimos la expresión (3.1) en la (3.2):

$$Q_\pi(s,a) = \sum_{s'\in\mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma \sum_{a'\in\mathcal{A}} \pi(a'|s') Q_\pi(s',a') \right) \tag{3.4}$$

3.4. Estrategia óptima

Como hemos comentado al principio de este apartado, el objetivo principal en desarrollar las ecuaciones de Bellman es encontrar una estrategia óptima para cada estado, es decir, aquella que maximiza las funciones valor sobre todas las estrategias π.

Definición 3.1. *Definimos las* ***funciones valor óptimas*** *como* $V_*(s) = \max_\pi V_\pi(s)$ *para todo* $s \in \mathcal{S}$ *y* $Q_*(s,a) = \max_\pi Q_\pi(s,a)$ *para todo* $(s,a) \in \mathcal{S}\times\mathcal{A}$.

Podemos definir la estrategia óptima como aquella que lleva a conseguir que la función valor sea óptima. Vamos a enunciar esta idea más formalmente con el siguiente teorema.

Teorema 3.2. *Si definimos un orden parcial sobre las estrategias tal que* $\pi \geq \pi'$ *si* $V_\pi(s) \geq V'_\pi(s)$ *para todo* $s \in \mathcal{S}$*, entonces para cualquier MDP se cumplen:*

1. *Existe una estrategia óptima* π^* *que es mejor o igual que el resto de estrategias del espacio de estrategias* Π*, es decir,* $\exists\pi^* \in \Pi$ *tal que* $\pi^* \geq \pi$ *para todo* $\pi \in \Pi$.
2. *Toda estrategia óptima logra que la función valor del estado y la de acción sea óptima, es decir,* $V_{\pi^*}(s) = V_*(s)$ *y* $Q_{\pi^*}(s,a) = Q_*(s,a)$.

La idea y enfoque principales son encontrar primero la función valor óptima y partir de esta, extraer la estrategia óptima. Es por eso que más adelante presentaremos diferentes algoritmos para encontrar V_* y Q_*, y a continuación derivar la estrategia óptima a partir de ellas. Formulamos esta idea en el siguiente lema.

Lema 3.3. *La estrategia óptima puede ser encontrada mediante la maximización de la acción* $a \in \mathcal{A}$ *sobre la función valor* $Q_*(s,a)$ *óptima:*

$$\pi^*(a|s) = \begin{cases} 1 & si\ a = arg\max_{a\in\mathcal{A}} Q_*(s,a) \\ 0 & en\ caso\ contrario \end{cases}$$

Antes de desarrollar las ecuaciones óptimas de Bellman para las funciones valor, introduciremos el concepto de *greediness* o *avaricia*.

Definición 3.4. *Diremos que una estrategia es* ***greedy*** *para algún estado $s \in \mathcal{S}$ respecto a una función valor $V_\pi : S \longrightarrow \mathbb{R}$ o $Q_\pi : \mathcal{S} \times \mathcal{A}_s \longrightarrow \mathbb{R}$, si esta estrategia maximiza la función valor:*

$$\pi(s) = arg\max_{a \in \mathcal{A}} Q(s,a) = arg\max_{a \in \mathcal{A}} \left(\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} (\mathcal{R}^a_{ss'} + \gamma V_\pi(s')) \right)$$

y análogamente para $V(s)$.

Con esto, observamos directamente que la estrategia óptima π^* será *greedy* respecto a la función valor óptima. Sabiendo esto, a continuación desarrollamos las ecuaciones óptimas.

Ecuación óptima de Bellman para V

Hemos visto anteriormente que la función V depende de la estrategia, es decir, que el valor de estado varia en función de la estrategia que sigue el agente, y por tanto, podemos tener muchas funciones valor diferentes según todas las estrategias posibles. La función V_* óptima es la que produce el valor máximo en comparación con todas las demás funciones valor de estado:

$$V_*(s) = \max_\pi V_\pi(s)$$

Podemos calcular la ecuación óptima de Bellman seleccionando la acción que da el valor máximo. Entonces, en lugar de usar una estrategia cualquiera π para seleccionar una acción, calculamos el valor del estado usando todas la acciones posibles y luego seleccionamos el valor máximo como el valor del estado. Tomamos la expresión de la función V en (3.3). Como no estamos usando ninguna estrategia, podemos eliminar la esperanza matemática sobre la estrategia π, tomar el máximo sobre la acción y así expresar la **ecuación óptima de Bellman para la función V** como:

$$V_*(s) = \max_a \left(\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} (\mathcal{R}^a_{ss'} + \gamma V_*(s')) \right) \tag{3.5}$$

Ecuación óptima de Bellman para Q

Igual que hemos hecho para la función V, podemos encontrar equivalentemente la función Q óptima:

$$Q_*(s,a) = \max_\pi Q_\pi(s,a)$$

Del mismo modo que hemos hecho para calcular la ecuación óptima de Bellman para la función V, en lugar de usar la estrategia para seleccionar la acción a' en el siguiente estado s', elegimos todas las acciones posibles en ese estado s' y calculamos el valor de la función Q máximo. Por tanto, partiendo de la expresión (3.4), expresamos la **ecuación óptima de Bellman para la función Q** como:

$$Q_*(s,a) = \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q_*(s', a') \right) \tag{3.6}$$

Finalmente nos queda añadir el hecho de que el valor óptimo de un estado, $V_*(s)$, es igual a la mejor función valor de la acción que podamos obtener a partir de este estado, es decir:

$$V_*(s) = \max_a Q_*(s, a)$$

Demostración. Suponemos que $V_*(s) = \max_a Q_*(s, a)$ y usando las expresiones en (3.5) y (3.6) veremos que es cierto.

Sabemos entonces que se cumple $V_*(s') = \max_{a'} Q_*(s', a')$. Tomamos la ecuación óptima de Bellman para V y directamente vemos que:

$$\begin{aligned} V_*(s) &= \max_a \left(\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} (\mathcal{R}^a_{ss'} + \gamma V_*(s')) \right) \\ &= \max_a \left(\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} (\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q_*(s', a')) \right) \\ &= \max_a Q_*(s, a) \end{aligned}$$

□

De esta forma, lo que buscamos al resolver un Proceso de Decisión Markoviano es encontrar la solución a las ecuaciones óptimas de Bellman que, como hemos dicho, equivale a encontrar la estrategia óptima. Veremos a continuación que existen dos enfoques básicos: cuando el modelo es conocido (modelo determinista) o cuando el modelo es desconocido (modelo estocástico). En cada uno de estos enfoques usaremos métodos diferentes para encontrar la estrategia óptima.

4. Programación Dinámica Determinista

Los modelos deterministas son aquellos donde se supone que los datos se conocen con certeza, es decir, se supone que cuando el modelo sea analizado se tiene disponible toda la información necesaria para la toma de decisiones. En nuestro caso, usaremos programación dinámica determinista cuando, sabiendo el estado y la decisión que posee la etapa actual, podemos determinar por completo la siguiente etapa.

Los modelos de programación dinámica determinista (PDD) constituyen una clase importante de Procesos de Decisión Markovianos y que es ampliamente estudiada. Las aplicaciones incluyen encontrar la ruta más corta en una red o en un control de inventario con demandas conocidas.

En modelos PDD, cuando elegimos una acción $a \in \mathcal{A}$, ésta determina el siguiente estado $s' \in \mathcal{S}$ con certeza. Las formulaciones estándar tienen esto en cuenta mediante el uso de una *función de transición de estado determinista*, en lugar de una probabilidad de transición, para especificar el siguiente estado.

Definición 4.1. *Una* ***función de transición de estado determinista*** *es una función* $f(s, a) : \mathcal{S} \times \mathcal{A}_s \longrightarrow \mathcal{S}$ *que especifica el estado del sistema a tiempo* $t+1$ *cuando el agente elije la acción* $a \in \mathcal{A}_s$ *en el estado* s *a tiempo* t*.*

Para formular un modelo PDD como un Proceso de Decisión Markoviano, definimos la probabilidad de transición como:

$$\mathcal{P}^a_{ss'} = \begin{cases} 1 \text{ si } f(s,a) = s' \\ 0 \text{ si } f(s,a) \neq s' \end{cases}$$

4.1. Proceso a tiempo discreto

Antes de adentrarnos en buscar métodos analíticos y desarrollar ecuaciones explícitas para resolver el problema de encontrar una estrategia óptima, vamos a describir las bases cuando nos encontramos en el caso de programación dinámica determinista discreta:

- Asumimos que el tiempo evoluciona de forma discreta. Esto significa que $t \in \{0, 1, 2...\}$, es decir, $t \in \mathbb{N} \cup \{0\}$.
- En cada instante t tenemos dos variables: un estado s_t que observa el agente y la acción que realiza a_t.
- Después de realizar cada acción, el agente recibe una recompensa $R(s_t, a_t)$ que depende del estado del proceso y de la acción tomada.
- Sabemos el valor inicial de la variable estado s_0 y también sabemos cuál es la función de transición $f(s, a)$, que nos indica el siguiente estado dados el estado y la acción a tiempo t, $(f(s_t, a_t) = s_{t+1})$.
- Asumimos que tanto el conjunto de estados $\mathcal{S}$ como el conjunto de acciones $\mathcal{A}$ son conjuntos finitos.
- Para cualquier secuencia de acciones $a \equiv \{a_0, a_1, ...\}$ con $a_t \in \mathcal{A}$, el proceso puede derivar en una gran cantidad de caminos factibles, es decir, en muchas secuencias de estados $s \equiv \{s_0, s_1, ...\}$ donde $s_{t+1} = f(s_t, a_t)$ con $a_t \in \mathcal{A}$.
- La acción que toma el agente en cada estado viene dada por una estrategia π, y ésta da lugar a la función valor V, que nos da un criterio para evaluar todos los posibles recorridos que puede tomar nuestro problema.
- El objetivo es encontrar una estrategia óptima π^* que nos muestre cuál es la secuencia de acciones que debemos tomar para maximizar nuestra función valor: $a^* = \{a_0^*, a_1^*, ..\}$. Y estas acciones dan lugar al camino óptimo $s^* = \{s_0, s_1^*, s_2^*, ...\}$.

El Principio de Optimalidad se puede expresar formalmente como hemos visto anteriormente en términos de la función valor V. Para cada instante t y estado s_t, vimos que el principio de Bellman implica que las funciones valor deben satisfacer las ecuaciones de Bellman para V y Q (ecuaciones (3.3) y (3.4)).

La ecuación de Bellman capta el problema esencial al que nos enfrentamos cuando el agente es dinámico y su objetivo es optimizar las recompensas en el futuro: la necesidad de equilibrar la recompensa inmediata $R(s_t, a_t)$ con el valor presente esperado de las recompensas futuras. Dadas las funciones valor, las estrategias óptimas π^* simplemente son las soluciones a los problemas de optimización incorporados en la ecuación de Bellman.

Nos podemos encontrar con un modelo que tenga un horizonte infinito ($T = \infty$) o un horizonte finito ($T < \infty$). En cualquiera de los dos casos, el objetivo es maximizar el retorno, es decir la función valor:

$$\sum_{t=0}^{T} \gamma^t R(s_t, a_t), 0 < \gamma < 1$$

Definición 4.2. *El **problema de control óptimo** (optimal control problem, OCP) se define como el problema que trata de encontrar la estrategia óptima* $\{a_t^*, s_t\}_{t=0}^{\infty}$ *que solucione* $\max_{\{a_t\}_{t=0}^{\infty}} \sum_{t=0}^{T} \gamma^t R(s_t, a_t)$ *tal que* $a_t \in \mathcal{A}$ *y* $s_{t+1} = f(s_t, a_t)$*, dado el estado inicial* s_0*.*

Uno de los métodos para resolver este tipo de problemas es a través del principio de la programación dinámica (Principio de Optimalidad de Bellman), que recordamos decía: "*Dado el estado y la acción iniciales, y la acción al comienzo de cualquier instante, las decisiones siguientes constituyen una estrategia óptima con respecto al estado resultante*". Siguiendo este principio, y según el tipo de horizonte, vamos a ver cómo resolvemos el problema.

4.1.1. Horizonte finito

En un modelo de horizonte finito, adoptamos el convenio de que el agente tiene que tomar una decisión para cada instante t, incluyendo la decisión final en la instante $T < \infty$. El agente no se enfrenta a decisiones después de t, pero sí que recibe una recompensa final $V_{T+1}(s_{T+1})$ en el periodo posterior que depende de la realización del estado en ese periodo. En muchas aplicaciones $V_{T+1}(s_{T+1})$ suele ser idénticamente cero, indicando que no se reciben recompensas después del instante de decisión final.

Vamos a suponer que conocemos la solución al problema de control óptimo $\{a_t^*, s_t\}_{t=0}^{T}$. En este caso, podemos definir la función valor del estado a tiempo τ como:

$$V_{T-\tau}(s_\tau) = \sum_{t=\tau}^{T} \gamma^{t-\tau} R(s_\tau, a_\tau^*)$$

Entonces, para $\tau = 0$ obtenemos

$$\begin{aligned} V_T(s_0) &= \max_{\{a_t\}_{t=0}^{T}} \sum_{t=0}^{T} \gamma^t R(s_t, a_t) \\ &= \max_{\{a_t\}_{t=0}^{T}} \left(R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + ... \right) \\ &= \max_{\{a_t\}_{t=0}^{T}} \left(R(s_0, a_0) + \gamma \sum_{t=1}^{T} \gamma^{t-1} R(s_t, a_t) \right) \end{aligned}$$

Aplicando el principio de optimalidad,

$$V_T(s_0) = \max_{a_0} \left(R(s_0, a_0) + \gamma \max_{\{a_t\}_{t=1}^{T}} \sum_{t=1}^{T} \gamma^{t-1} R(s_t, a_t) \right)$$

Y por la definición de $V_{T-\tau}(s_\tau)$, con $\tau = 1$, tenemos que

$$V_T(s_0) = \max_{a_0} \left(R(s_0, a_0) + \gamma V_{T-1}(s_1)\right)$$

Iterando, para cualquier instante $0 \leq t \leq T$, encontramos la **Ecuación de Hamilton-Jacobi-Bellman (HJB)**:

$$V_{T-t}(s_t) = \max_{a_t} \left(R(s_t, a_t) + \gamma V_{T-t-1}(s_{t+1})\right) \tag{4.1}$$

Vemos que esta misma ecuación la podemos obtener de la ecuación de Bellman óptima para V, en (3.5). La diferencia es que ahora estamos en un proceso discreto, por tanto, la probabilidad de transición es 1. Y además, al estar en un proceso determinista, sabiendo el estado y acción actual, el siguiente estado está totalmente determinado. Es por esto que nos queda

$$V_*(s_t) = \max_a \left(\sum_{s_{t+1} \in \mathcal{S}} \mathcal{P}^a_{s_t s_{t+1}} \left(\mathcal{R}^a_{s_t s_{t+1}} + \gamma V_*(s_{t+1}) \right) \right) \Rightarrow$$

$$\Rightarrow V_{T-t}(s_t) = \max_{a_t} \left(R(s_t, a_t) + \gamma V_{T-t-1}(s_{t+1})\right)$$

Habiendo encontrado la ecuación de HJB (4.1), tenemos la siguiente proposición:

Proposición 4.3. *Dada una solución óptima al problema de control óptimo, entonces ésta verifica la ecuación de HJB.*

Entonces, resolvemos el problema de control óptimo a través de la recursión que obtenemos directamente de la ecuación de HJB (donde f es la función de transición)

$$V_{t+1}(s) = \max_a \left(R(s, a) + \gamma V_t(f(s, a))\right) \tag{4.2}$$

Dado el valor de V_0, se resuelve de forma recursiva hacia atrás, como se muestra a continuación:

- Teniendo V_0, resolvemos $V_1(s)$ para todos los estados s.
- Teniendo V_1, resolvemos $V_2(s)$ para todos los estados s.
- ... y así recursivamente.
- Hasta que finalmente obtenemos $V_T(s)$.

Como solo se pueden tomar un número de acciones finito, el problema de optimización que se deriva de las ecuaciones de Bellman siempre se puede resolver realizando un número finito de operaciones aritméticas. Además, las funciones valor en un Proceso de Decisión Markoviano discreto con horizonte finito siempre están bien definidas, aunque en algunos casos existe más de una estrategia que maximice el valor esperado de las recompensas, es decir, la acción óptima puede no ser única.

4.1.2. Horizonte infinito

Si el problema de decisión tiene un horizonte infinito, la función límite $V = \lim_{j\to\infty} V_j$ es independiente de j, es decir, la función valor no dependerá del tiempo t. Por tanto, escribiremos la ecuación de HJB como:

$$V(s) = \max_a \left(R(s,a) + \gamma V(f(s,a))\right) = \max_a H(s,a)$$

En este caso, para resolver la ecuación de HJB, y si se dan ciertas condiciones de regularidad para la función V, determinamos el control óptimo a través de la condición de optimalidad

$$\frac{\partial H(s,a)}{\partial a} = 0$$

Si H es C^2, damos con la **función de estrategia** $a^* = h(s)$, que nos da una norma o regla óptima para cambiar la acción óptima, dado el estado. Con esto, la ecuación de HJB se convierte en una EDO no-lineal:

$$V(s) = R(s,h(s)) + \gamma V[f(s,h(s))] \tag{4.3}$$

4.1.3. Ejemplo

Referencia: *Introduction to Dynamic Programming Applied to Economics* (2007), Paulo Brito.

A continuación vamos a exponer un ejemplo de un problema (el problema del consumidor) para el caso determinista discreto y de cómo resolverlo usando programación dinámica determinista y la ecuación de HJB que hemos desarrollado.

Asumimos que tenemos $T > 1$ instantes y que los consumidores son homogéneos. Los consumidores tienen una serie de dotaciones financieras o *endowments* (en inglés) $y \equiv \{y_t\}_{t=0}^T$, es decir, donaciones a una organización de dinero o propiedades, que utiliza los ingresos de inversión resultantes para un propósito específico. Conocemos $y \equiv \{y_t\}_{t=0}^T$ con certeza ya que estamos en un proceso determinista. Tenemos mercados al contado, es decir mercados en los cuales tanto la transacción como la liquidación de una operación coinciden en la misma fecha. Existe un mercado al contado de bienes y un mercado al contado para un activo financiero. El activo financiero tiene el derecho a recibir el dividendo D_t al final de cada instante t. Los precios al contado son Q_t para el bien y P_t para el activo financiero. Asumimos que el mercado de bienes abre al principio de cada instante y que el mercado del activo abre al final.

El problema del consumidor: Elegir una secuencia de consumo $\{c_t\}_{t=0}^T$ y una secuencia $\{\theta\}_{t=0}^T$, que representa la cantidad de activo comprado al principio de cada instante t. El objetivo del consumidor es encontrar

$$\max_{\{c_t,\theta_{t+1}\}_{t=0}^T} \sum_{t=0}^{T} \gamma^t R(c_t)$$

donde $R(c_t)$ denota la recompensa que recibe el consumidor a tiempo t por el consumo c_t, es decir, es la función de utilidad del consumidor. Suponemos que la función de recompen-

sa $R(c_t)$ es una función continua, diferenciable, creciente, cóncava y homogénea de grado n.

Además, si denotamos por A_t la riqueza financiera al comienzo del instante t, entonces A_{t+1} será igual a la cantidad de activo en $t+1$ por el precio del activo en el instante anterior: $A_{t+1} = \theta_{t+1} P_t$. Sabiendo que para cada instante t, el precio total de los activos comprados junto con el dividendo que se recibe por ellos es igual al consumo más el stock, obtenemos las siguientes restricciones:

$$\begin{aligned} A_0 + y_0 &= c_0 + \theta_1 P_0 \\ \theta_1(P_1 + D_1) + y_1 &= c_1 + \theta_2 P_1 \\ &\dots \\ \theta_t(P_t + D_t) + y_t &= c_t + \theta_{t+1} P_t \\ &\dots \\ \theta_T(P_T + D_T) + y_T &= c_T \end{aligned}$$

Entonces la **restricción presupuestaria** para cada t es

$$A_{t+1} = y_t - c_t + r_t A_t \tag{4.4}$$

para $t = 0, ..., T$ y con el retorno de activo $r_t = \frac{P_t + D_t}{P_{t-1}}$.

Demostración. Sabiendo que $A_{t+1} = \theta_{t+1} P_t$,

$$A_{t+1} = \theta_{t+1} P_t = y_t - c_t + \theta_t(P_t + D_t) = y_t - c_t + \theta_t P_{t-1} \frac{P_t + D_t}{P_{t-1}} = y_t - c_t + A_t r_t$$

□

Por tanto, desarrollando la función valor como vimos para el caso determinista discreto para el horizonte finito, la **ecuación de HJB** será

$$V(A_t) = \max_{c_t}(R(c_t) + \gamma V(A_{t+1}))$$

También la podemos escribir como

$$V(A) = \max_{c} \Big(R(c) + \gamma V(\tilde{A}) \Big) \tag{4.5}$$

donde $\tilde{A} = y - c + rA$, entonces $V(\tilde{A}) = V(y - c + rA)$.
Para encontrar este máximo, derivamos en función de c e igualamos a 0:

$$R'(c) + \gamma V'(\tilde{A})(-1) = 0$$

Si suponemos que la solución óptima para el problema del consumidor es $\{c_t^*\}_{t=0}^T$, entonces tenemos la **condición de optimalidad**

$$R'(c^*) = \gamma V'(\tilde{A})$$

Si pudiéramos encontrar la función de estrategia $c^* = h(A)$ y sustituirla en la ecuación de HJB, obtendríamos

$$V(A) = R(h(A)) + \gamma V(\tilde{A}^*) \text{ con } \tilde{A}^* = y - h(A) + rA$$

Nuestro objetivo principal es determinar la función valor óptima, es decir, encontrar una solución explícita a la ecuación de HJB (4.5). Tenemos que determinar conjuntamente la función de estrategia $h(A)$ y la función valor óptima $V(A)$. Para esto vamos a usar el ejemplo en que la recompensa es una función logarítmica $R(c) = \log(c)$, y para simplificar el problema vamos a suponer que $y \equiv 0$. La solución es:

$$\text{Función valor óptima: } \boxed{V(A) = (1-\gamma)^{-1} \log\left((r\Theta)^{(1-\gamma)^{-1}} A\right)}, \text{ con } \Theta \equiv (1-\gamma)^{1-\gamma}\gamma^{\gamma}$$

$$\text{Consumo óptimo: } \boxed{c^* = h(A) = (1-\gamma)rA}$$

Demostración. Con $R(c) = \log(c)$, la condición de optimalidad es $\frac{1}{c^*} = \gamma V'(-c^* + rA) \Rightarrow c^* = [\gamma V'(-c^* + rA)]^{-1}$.
Para la función valor usaremos el método de los coeficientes indeterminados. Asumimos que la función valor es de la forma

$$V(A) = B_0 + B_1 \log(A) \tag{4.6}$$

donde B_0 y B_1 son los coeficientes indeterminados. Tenemos que determinar estos coeficientes como funciones de los parámetros de la ecuación de HJB. Si aplicamos

$$V'(A) = \frac{B_1}{A}$$

a la condición de optimalidad, obtenemos

$$c^* = \left[\gamma \frac{B_1}{-c^* + rA}\right]^{-1} = \frac{-c^* + rA}{\gamma B_1} = \frac{rA}{1 + \gamma B_1} \tag{4.7}$$

Entonces encontramos la siguiente función lineal de A:

$$\tilde{A}^* = -c^* + rA = -\frac{rA}{1+\gamma B_1} + rA = \left(\frac{\gamma B_1}{1+\gamma B_1}\right) rA$$

Si lo sustituimos en la ecuación de HJB

$$\begin{aligned} V(A) = B_0 + B_1 \log(A) = R(c^*) + \gamma V(\tilde{A}^*) &= \log\left(\frac{rA}{1+\gamma B_1}\right) + \gamma\left[B_0 + B_1 \log\left(\frac{\gamma B_1}{1+\gamma B_1} rA\right)\right] \\ &= \log\left(\frac{r}{1+\gamma B_1}\right) + \log(A) + \gamma\left[B_0 + B_1 \log\left(\frac{\gamma B_1 r}{1+\gamma B_1}\right) + \log(A)\right] \end{aligned}$$

Vemos que podemos eliminar el término $\log(A)$ si $B_1 = 1 + \gamma B_1$, es decir si

$$B_1 = \frac{1}{1-\gamma}$$

Así, la ecuación de $V(A)$ se reduce a

$$B_0(1-\gamma) = \log((1-\gamma)r) + \frac{\gamma}{1-\gamma}\log(r\gamma)$$

que podemos resolver para B_0.

$$\begin{aligned} B_0 = \frac{\log((1-\gamma)r)}{1-\gamma} + \frac{\gamma}{(1-\gamma)^2}\log(r\gamma) = \frac{(1-\gamma)\log((1-\gamma)r) + \gamma\log(r\gamma)}{(1-\gamma)^2} = \\ = \frac{\log((1-\gamma)^{1-\gamma} r^{1-\gamma} r^{\gamma}\gamma^{\gamma})}{(1-\gamma)^2} = (1-\gamma)^{-2}\log(r\Theta) \text{ , con } \Theta \equiv (1-\gamma)^{1-\gamma}\gamma^{\gamma} \end{aligned}$$

Finalmente, sustituyendo las funciones que hemos encontrado, B_0 y B_1, en (4.6) y en (4.7), escribimos las expresiones de la función valor óptima y del consumo óptimo, respectivamente. □

4.2. Proceso a tiempo continuo

Nos encontramos en un proceso ahora donde el tiempo evoluciona de forma continua. Esto significa que $\mathcal{T} = \mathbb{R}_+$. En este tipo de procesos volveremos a distinguir dos casos, cuando el horizonte es finito, es decir hay un tiempo final T y que es el problema más simple, y cuando el horizonte es infinito.

4.2.1. Horizonte finito

Ahora nos encontramos en un espacio de funciones $(s(t), a(t))$, con $t_0 \leq t \leq T$. Nuestro objetivo está en encontrar funciones $(s^*(t), a^*(t))$ que resuelvan el siguiente problema:

$$\max_{a(t)} \int_{t_0}^{T} R(t, s(t), a(t))dt$$

teniendo en cuenta que

$$\dot{s} \equiv \frac{ds(t)}{dt} = f(t, s(t), a(t))$$

dado el estado inicial $s(t_0) = s_0$. Observamos que $f(t, s(t), a(t))$ reemplaza a $s_{t+1} = f(s_t, a_t)$ que teníamos en el caso discreto. Asumimos que T es conocido. Recordamos que f es la función de transición de estado, que usamos en lugar de una probabilidad de transición.

Entonces, la función valor del estado, para el instante inicial es

$$\mathcal{V}(t_0, s_0) = \int_{t_0}^{T} R(t, s^*, a^*)dt$$

y para el último instante es

$$\mathcal{V}(T, s(T)) = 0$$

A continuación enunciamos un lema que nos da las condiciones necesarias para conseguir la optimalidad a partir del Principio de Optimalidad.

Lema 4.4. *Sea $\mathcal{V} \in \mathcal{C}^2(\mathcal{T}, \mathbb{R})$. La función valor asociada al camino óptimo $\{(s^*(t), a^*(t)) : t_0 \leq t \leq T\}$ verifica la ecuación en derivadas parciales no-lineal o **Ecuación de Hamilton-Jacobi-Bellman (HJB)**:*

$$-\mathcal{V}_t(t, s) = \max_a [R(t, s, a) + \mathcal{V}_s(t, s)f(t, s, a)] \tag{4.8}$$

donde $\mathcal{V}_t(t, s) := \frac{\partial \mathcal{V}(t,s)}{\partial t}$ y $\mathcal{V}_s(t, s) := \frac{\partial \mathcal{V}(t,s)}{\partial s}$.

Demostración. Consideramos la función valor

$$\mathcal{V}(t, s) = \max_a \left(\int_t^T R(u, s, a)du \right)$$

Queremos encontrar una ecuación diferencial parcial para $\mathcal{V}(t, s)$. Suponemos en todo momento que $R(\cdot)$ es continua como función total de t, con $s = s(t)$ y $a = a(t)$. Para todo

$h > 0$ podemos escribir

$$\begin{aligned}\mathcal{V}(t,s) &= \max_a \left(\int_t^{t+h} R(u,s,a)du + \int_{t+h}^{T} R(u,s,a)du \right) \\ &= \max_a \left[\int_t^{t+h} R(u,s,a)du + \max_a \left(\int_{t+h}^{T} R(u,s,a)du \right) \right] \\ &= \max_a \left[\int_t^{t+h} R(u,s,a)du + \mathcal{V}(t+h, s_{t+h}) \right]\end{aligned}$$

Para encontrar la segunda igualdad usamos el principio de la programación dinámica. Asumiendo que $\mathcal{V}$ es $\mathcal{C}^2(\mathcal{T}, \mathbb{R})$, vemos que

$$\mathcal{V}(t+h, s_{t+h}) = \mathcal{V}(t,s) + \int_t^{t+h} \left(\frac{\partial \mathcal{V}}{\partial u} + \frac{\partial \mathcal{V}}{\partial s} f(u,s,a) \right) du$$

Como consecuencia

$$\mathcal{V}(t,s) = \mathcal{V}(t,s) + \max_a \left(\int_t^{t+h} R(u,s,a)du + \int_t^{t+h} (\mathcal{V}_u + \mathcal{V}_s f(u,s,a))du \right)$$

y

$$\begin{aligned}0 &= \max_a \left(\int_t^{t+h} R(u,s,a)du + \int_t^{t+h} (\mathcal{V}_u + \mathcal{V}_s f(u,s,a))du \right) \\ &= \max \int_t^{t+h} \left(\mathcal{V}_u + \max_a (R(u,s,a) + \mathcal{V}_s f(u,s,a)) \right) du\end{aligned}$$

Como esto es cierto para todo $h > 0$, obtenemos que

$$\mathcal{V}_t + \max_a (R(t,s,a) + \mathcal{V}_s f(t,s,a)) = 0$$

□

Igual que en el caso discreto, en el caso continuo llamamos a $a^* = h(t,s)$ la **función de estrategia**. Entonces, la ecuación de HJB se puede escribir como:

$$-\mathcal{V}_t(t,s) = R(t,s,h(t,s)) + \mathcal{V}_s(t,s) f(t,s,h(t,s)) \tag{4.9}$$

Aunque la diferenciabilidad de $\mathcal{V}$ está asegurada para las funciones R y f, podemos encontrar soluciones explícitas para $\mathcal{V}(\cdot)$ y $h(\cdot)$ en casos muy puntuales, pero no suele ocurrir.

4.2.2. Horizonte infinito

Nuestro objetivo ahora es el mismo que antes, pero a diferencia del caso anterior tenemos que $t \geq t_0$. Queremos entonces encontrar funciones $(s^*(t), a^*(t))$ que resuelvan el siguiente problema:

$$\max_{a(t)} \int_{t_0}^{\infty} R(s(t), a(t)) e^{-\rho t} dt$$

donde $\rho > 0$ y teniendo en cuenta que

$$\dot{s}(t) \equiv \frac{ds}{dt} = f(s(t), a(t)), t \geq t_0$$

dado el estado inicial $s(t_0) = s_0$.

El siguiente lema, similar al anterior, nos da las condiciones necesarias para conseguir la optimalidad en este caso.

Lema 4.5. *Sea $\mathcal{V} \in C^2(\mathbb{T}, \mathbb{R})$. La función valor asociada al camino óptimo $\{(s^*(t), a^*(t)) : t_0 \leq t < \infty\}$ verifica la EDO no-lineal fundamental o* ***Ecuación de Hamilton-Jacobi-Bellman (HJB):***

$$\rho V(s) = \max_a \left[R(s,a) + V'(s) f(s,a)\right] \tag{4.10}$$

Demostración. Consideramos la función valor

$$\begin{aligned}\mathcal{V}(t_0, s_0) &= \max_a \left(\int_{t_0}^{\infty} R(s,a) e^{-\rho t} dt\right) \\ &= e^{-\rho t_0} \max_a \left(\int_{t_0}^{\infty} R(s,a) e^{-\rho(t-t_0)} dt\right) \\ &= e^{-\rho t_0} V(s_0)\end{aligned}$$

donde $V(\cdot)$ es independiente de t_0 y solamente depende de s_0. Podemos hacer

$$V(s_0) = \max_a \left(\int_0^{\infty} R(s,a) e^{-\rho t} dt\right)$$

Entones obtenemos, para cada (t, s), que $\mathcal{V}(t, s) = e^{-\rho t} V(s)$. Si calculamos las derivadas,

- $\mathcal{V}_t(t,s) \equiv \frac{\partial \mathcal{V}(t,s)}{\partial t} = -\rho e^{-\rho t} V(s)$
- $\mathcal{V}_s(t,s) \equiv \frac{\partial \mathcal{V}(t,s)}{\partial s} = e^{-\rho t} V'(s)$

y sustituimos en la ecuación de HJB (4.8) que hemos obtenido para el horizonte finito, vemos que

$$\begin{aligned}\rho e^{-\rho t} V(s) = \max_a \left[R(s,a) e^{-\rho t} + e^{-\rho t} V'(s) f(s,a)\right] \Rightarrow \\ \Rightarrow \rho V(s) = \max_a \left[R(s,a) + V'(s) f(s,a)\right]\end{aligned}$$

□

Si determinamos la función de estrategia $a^* = h(s)$ y substituimos en la ecuación de HJB en (4.9), obtenemos

$$\rho V(s) = R(s, h(s)) + V'(s) f(s, h(s)) \tag{4.11}$$

Vemos que es una EDO del tipo $\dot{x}(t) = a(t) + b(t)x(t)$, por tanto, esta nueva ecuación de HJB se puede definir como una recursión sobre s. Igual que en el caso anterior, en este es también poco común encontrar soluciones explícitas para $V(s)$.

4.2.3. Ejemplo

Referencia: *Introduction to Dynamic Programming Applied to Economics* (2007), Paulo Brito.

Ahora vamos a exponer el mismo problema que hicimos para el proceso discreto, el problema del consumidor, pero ahora para el caso determinista continuo, usando también programación dinámica y la ecuación de HJB.

Asumimos que $\mathcal{T} = \mathbb{R}_+$, es decir, las decisiones y transacciones tienen lugar de forma continua en el tiempo. Asumimos también que los consumidores son homogéneos, es decir, tienen las mismas dotaciones financieras y las mismas preferencias. Como estamos en un entorno determinista, los consumidores tienen la información completa sobre el flujo de dotaciones financieras, que ahora para el caso continuo son $y \equiv \{y(t), t \in \mathbb{R}_+\}$. Existen mercados al contado que están continuamente abiertos, donde $Q(t)$ indica el precio. Además hay un mercado de activos en el que se negocia un solo activo con precio $P(t)$ y se paga un dividendo $D(t)$.

Problema del consumidor: Encontrar una función óptima de consumo $c(t)$, con tal de maximizar la función valor:

$$\mathcal{V}(c(t)) = \int_0^\infty R(c(t))e^{-\rho t}dt$$

Suponemos que la función de recompensa $R(c(t))$ es continua, diferenciable, creciente, cóncava y homogénea de grado n. El consumidor elige el número de activos $\theta(t)$. Si consideramos un pequeño incremento en el tiempo que llamamos h y asumimos que el flujo de variables es constante en ese intervalo de tiempo, entonces, de forma similar al caso discreto, tenemos:

$$P(t+h)\theta(t+h) = \theta(t)P(t) + \theta(t)D(t)h + Q(t)(y(t) - c(t))h + O(h)$$

Si definimos $A(t)$ como el stock de riqueza financiera en t, entonces $A(t) = P(t)\theta(t)$. Por tanto, la ecuación anterior es equivalente a

$$A(t+h) = A(t) + \frac{A(t)D(t)}{P(t)}h + Q(t)(y(t) - c(t))h + O(h)$$

Si denotamos por i a la tasa de retorno nominal, $i(t) = \frac{D(t)}{P(t)}$, podemos escribir

$$A(t+h) - A(t) = i(t)A(t)h + Q(t)(y(t) - c(t))h + O(h)$$

Dividiendo por h y tomando el límite cuando $h \to 0$, obtenemos

$$\frac{dA(t)}{dt} = i(t)A(t) + Q(t)(y(t) - c(t))$$

Definimos el conjunto de riqueza como $a(t) \equiv \frac{A(t)}{Q(t)}$ y la **tasa de interés real** como $r(t) = i(t) - \frac{\dot{Q}}{Q(t)}$, con $\dot{Q} = \frac{dQ(t)}{dt}$.

Entonces obtenemos la restricción presupuestaria instantánea

$$\dot{a}(t) = r(t)a(t) + y(t) - c(t) \tag{4.12}$$

donde asumimos que sabemos $a(0) = a_0$.

Definimos también:

- **Riqueza conjunta** como $h(t) = \int_t^\infty e^{-\int_t^s r(\tau)d\tau} y(s)ds$.

Usando la fórmula de Leibniz para integrales, calculamos

$$\dot{h}(t) = \frac{dh(t)}{dt} = r(t)\int_t^\infty e^{-\int_t^s r(\tau)d\tau} y(s)ds - y(t)$$

$$\dot{h}(t) = r(t)h(t) - y(t) \tag{4.13}$$

Con esto, la **riqueza total** en el instante t es la suma de la riqueza $a(t)$ más la riqueza conjunta: $w(t) := a(t) + h(t)$. Entonces podemos expresar la restricción presupuestaria como una función de $w(\cdot)$,

$$\dot{w} = \dot{a}(t) + \dot{h}(t) = r(t)w(t) - c(t)$$

usando las expresiones en (4.12) y (4.13).

El problema del consumidor trata de encontrar $(c^*(t), w^*(t))$, que hace referencia a la función de estrategia óptima y a la acumulación óptima de riqueza, respectivamente, para $t \in \mathbb{R}_+$ que maximicen

$$\mathcal{V}(c(t)) = \int_0^\infty R(c(t))e^{-\rho t}dt$$

sujeto a la restricción presupuestaria instantánea

$$\dot{w} = \dot{a}(t) + \dot{h}(t) = r(t)w(t) - c(t) \text{ dado } w(0) = w_0$$

Por tanto, desarrollando la función valor como vimos para el caso determinista continuo, la **ecuación de HJB** será

$$\rho V(w) = \max_c \left\{R(c) + V'(w)(rw - c)\right\} \tag{4.14}$$

donde $w = w(t)$, $c = c(t)$, $r = r(t)$. Para encontrar este máximo, derivamos con respecto a c e igualamos a 0

$$R'(c) + V'(w)(-1) = 0$$

Si suponemos que la estrategia óptima es $c^*(t)$, entonces tenemos la **condición de optimalidad**

$$R'(c^*) = V'(w)$$

Hemos asumido que la función de recompensa $R(c)$ es homogénea de grado n, por tanto, tiene las siguientes propiedades:

$$R(c) = c^n R(1)$$
$$R'(c) = c^{n-1}R'(1)$$

Con la condición de optimalidad, damos con las soluciones a nuestro problema:

$$\text{Función de estrategia óptima: } \boxed{c^*(t) = \pi(t)w(t)}$$

$$\text{Acumulación óptima de riqueza: } \boxed{w^*(t) = w_0 e^{\int_0^t (r(s)-\pi(s))ds}}$$

con $\pi(t) = \left(\frac{nB(t)}{R'(1)}\right)^{\frac{1}{n-1}}$.

Demostración. A partir de la condición de optimalidad y las propiedades de $R(c)$, vemos que

$$R'(c^*) = (c^*)^{n-1}R'(1) = V'(w) \Rightarrow c^* = \left(\frac{V'(w)}{R'(1)}\right)^{\frac{1}{n-1}}$$

Sustituyendo en la ecuación de HJB 4.14 obtenemos

$$\begin{aligned}
\rho V(w) &= (c^*)^n R(1) + V'(w)rw - V'(w)c^* = \\
&= V'(w)rw + (c^*)^n R(1) - R'(c^*)c^* = \\
&= V'(w)rw + (c^*)^n R(1) - (c^*)^{n-1}R'(1)c^* = \\
&= V'(w)rw + (c^*)^n (R(1) - R'(1)) = \\
&= V'(w)rw + (R(1) - R'(1)) \left(\frac{V'(w)}{R'(1)}\right)^{\frac{1}{n-1}}
\end{aligned}$$

Vemos que esto es una EDO definida para $V(w)$. Para resolverla y encontrar la expresión de la función valor, suponemos que solución es de la forma: $V(w) = Bw^n$. Sustituimos en la ecuación anterior sabiendo que $V'(w) = nBw^{n-1}$.

$$\rho Bw^n = nBw^n r + (R(1) - R'(1)) \left(\frac{nBw}{R'(1)}\right)^n$$

Podemos eliminar el término w^n y nos queda

$$\begin{aligned}
\rho B &= nBr + (R(1) - R'(1)) \left(\frac{nB}{R'(1)}\right)^n \\
(\rho - nr)B &= (R(1) - R'(1)) \left(\frac{nB}{R'(1)}\right)^n \\
B^{1-n} &= \left(\frac{R(1) - R'(1)}{\rho - nr}\right) \left(\frac{n}{R'(1)}\right)^n \\
B &= \left[\left(\frac{R(1) - R'(1)}{\rho - nr}\right) \left(\frac{n}{R'(1)}\right)^n\right]^{\frac{1}{1-n}}
\end{aligned}$$

Entonces, como B es una función de $r = r(t)$, podemos determinar explícitamente la función valor

$$V(w(t)) = B(t)w(t)^n$$

A partir de la función valor, calculamos la función de estrategia

$$c^*(t) = \left(\frac{nB(t)}{R'(1)}\right)^{\frac{1}{n-1}} w(t) \equiv \pi(t)w(t)$$

Finalmente obtenemos la acumulación óptima de riqueza $w^*(t)$ como la solución de la EDO: $\dot{w}^* = r(t)w^*(t) - c^*(t) = r(t)w^*(t) - \pi(t)w^*(t) = (r(t) - \pi(t))w^*(t)$. □

5. Programación Dinámica Estocástica

En la sección anterior hemos analizado el caso en el que conocemos el estado de la siguiente etapa sabiendo el estado y decisión de la etapa actual. Si no sabemos con seguridad cuál será el siguiente estado y, en cambio, tenemos alguna función de distribución se habla de programación dinámica estocástica. Las ideas básicas de determinar los estados, las etapas, las estrategias y las ecuaciones funcionales siguen valiendo, pero toman una forma distinta. La aleatoriedad que aparece nos introduce a los procesos estocásticos. Es por esto que antes de desarrollar los métodos de programación dinámica primero daremos la definición de conceptos como la esperanza condicionada, los procesos estocásticos y las filtraciones.

5.1. Esperanza condicionada

Sea $(\Omega, \mathcal{F}, P)$ un espacio de probabilidad y consideremos dos variables aleatorias discretas X, Y entonces,

$$\mathbb{E}(X|Y=y) := \sum_x xP(X=x|Y=y) \quad \text{(suponemos } P(Y=y)>0)$$

y podemos definir, a partir de aquí, la variable

$$\mathbb{E}(X|Y)(\omega) = \mathbb{E}\left(X|Y=y\right), \text{ si } Y(\omega)=y$$

o lo que es lo mismo

$$\mathbb{E}(X|Y) = \sum_y \mathbb{E}(X|Y=y)\mathbf{1}_{\{Y=y\}}$$

Sea ahora $A \in \mathcal{F}$ (suponemos $P(A)>0$)

$$\mathbb{E}(X|A) := \sum_x xP(X=x|A),$$

consideremos la σ-álgebra generada por un partición, (A_i), de Ω, llamémosle $\mathcal{A}$, esto es $\mathcal{A} = \{\emptyset, \Omega, \cup_j A_{i_j}\}$, podemos ahora definir la variable aleatoria

$$\mathbb{E}(X|\mathcal{A})(\omega) = \mathbb{E}(X|A_i), \text{ si } \omega \in A_i,$$

o equivalentemente

$$\mathbb{E}(X|\mathcal{A}) = \sum_i \mathbb{E}(X|A_i)\mathbf{1}_{A_i}.$$

Notemos que $\mathbb{E}(X|\mathcal{A})$ es una variable aleatoria $\mathcal{A}$-medible:

$$\mathbb{E}(X|\mathcal{A})^{-1}(z_i) = A_i \in \mathcal{A} \text{ si } z_i = \mathbb{E}(X|A_i),$$

además para todo $A \in \mathcal{A}$

$$\mathbb{E}(X\mathbf{1}_A) = \mathbb{E}\left(\mathbb{E}(X|\mathcal{A})\mathbf{1}_A\right).$$

Demostración. En efecto, si tomamos $A = A_i$

$$\mathbb{E}(X|\mathcal{A})\mathbf{1}_A = \mathbb{E}(X|A_i)\mathbf{1}_{A_i},$$

de manera que

$$\begin{aligned}\mathbb{E}\left(\mathbb{E}(X|\mathcal{A})\mathbf{1}_A\right) &= \mathbb{E}\left(\mathbb{E}(X|A_i)\mathbf{1}_{A_i}\right) = \mathbb{E}(X|A_i)P(A_i)\\ &= \sum_x xP(X=x|A_i)P(A_i)\\ &= \sum_x xP(X=x,A_i) = \mathbb{E}(X\mathbf{1}_{A_i}).\end{aligned}$$

□

Esto nos lleva a la siguiente definición general.

Definición 5.1. *Consideremos una variable aleatoria integrable X y sea $\mathcal{A}$ una σ-álgebra contenida en $\mathcal{F}$, definimos la* ***esperanza condicionada de*** *X* ***sobre*** *$\mathcal{A}$ a un variable aleatoria, digamos Z, con las siguientes propiedades:*

(i) Z es $\mathcal{A}$-medible
(ii) Para todo $A \in \mathcal{A}$, $\mathbb{E}(Z\mathbf{1}_A) = \mathbb{E}(X\mathbf{1}_A)$.

Se puede demostrar que tal variable siempre existe y es única casi seguramente.

Vemos a continuación algunas de las principales propiedades de la esperanza condicionada.

Propiedades

1. Linealidad: $\mathbb{E}(aX+bY|\mathcal{A}) = a\mathbb{E}(X|\mathcal{A}) + b\mathbb{E}(Y|\mathcal{A})$, $a,b \in \mathbb{R}$.
2. $\mathbb{E}(\mathbb{E}(X|\mathcal{A})) = \mathbb{E}(X)$. Basta con tomar $A = \Omega$ en (ii).
3. Si X y $\mathcal{A}$ son independientes $\mathbb{E}(X|\mathcal{A}) = \mathbb{E}(X)$. De hecho la constante $\mathbb{E}(X)$ es trivialmente $\mathcal{A}$-medible, y para todo $A \in \mathcal{A}$, tenemos $\mathbb{E}(X\mathbf{1}_A) = \mathbb{E}(X)\mathbb{E}(\mathbf{1}_A) = \mathbb{E}(\mathbb{E}(X)\mathbf{1}_A)$.
4. Si X es $\mathcal{A}$-medible, entonces $\mathbb{E}(X|\mathcal{A}) = X$. Trivial a partir de la definición.
5. Si Y esta acotada y es $\mathcal{A}$-medible, entonces $\mathbb{E}(YX|\mathcal{A}) = Y\mathbb{E}(X|\mathcal{A})$. De hecho la variable aleatoria $Y\mathbb{E}(X|\mathcal{A})$ es integrable y $\mathcal{A}$-medible, y para todo $A \in \mathcal{A}$ tenemos
$$\mathbb{E}(\mathbb{E}(X|\mathcal{A})Y\mathbf{1}_A) = \mathbb{E}(XY\mathbf{1}_A),$$
donde la igualdad se deduce de (ii) y del teorema de convergencia dominada. Esta propiedad significa que las variables $\mathcal{A}$-medibles se comportan como constantes y pueden ser factorizadas fuera de la esperanza condicionada con respecto a $\mathcal{A}$. Esta propiedad es cierta siempre que $\mathbb{E}(|XY|) < \infty$. En particular si $X, Y \in L^2(\Omega)$.
6. En particular si $X, Y \in L^2(\Omega)$, por la propiedad anterior con $A = \Omega$, tenemos que si Y es $\mathcal{A}$-medible $X - \mathbb{E}(X|\mathcal{A})$ es ortogonal a Y :
$$\mathbb{E}((X - \mathbb{E}(X|\mathcal{A}))\, Y) = \mathbb{E}(XY) - (\mathbb{E}(X|\mathcal{A})Y) = 0.$$
7. (Propiedad iterativa) Dadas dos σ-álgebras $\mathcal{B} \subset \mathcal{A}$, entonces $\mathbb{E}(\mathbb{E}(X|\mathcal{A})|\mathcal{B}) = \mathbb{E}(X|\mathcal{B})$. En efecto, sea $B \in \mathcal{B}$, tenemos que ver que
$$\mathbb{E}(\mathbb{E}(X|\mathcal{A})\mathbf{1}_B) = \mathbb{E}(X\mathbf{1}_B),$$
pero como $\mathcal{B} \subset \mathcal{A}$, entonces $B \in \mathcal{A}$ y la igualdad es cierta por definición de $\mathbb{E}(X|\mathcal{A})$.

5.2. Procesos estocásticos y espacios filtrados

Sea un espacio de probabilidad $(\Omega, \mathcal{F}, P)$.

Definición 5.2. *Un* ***proceso estocástico*** X *es una familia de variables aleatorias* $\{X_t, t \geq 0\}$*, donde* $X_t : \Omega \to \mathbb{R}$*, definidos en el mismo espacio de probabilidad* $(\Omega, \mathcal{F}, P)$*.*

Definición 5.3. *Una* ***filtración*** $(\mathcal{F}_t)_{t\geq 0}$*, es una sucesión creciente de sub-*σ *álgebras de* $\mathcal{F}$*. El espacio* $(\Omega, (\mathcal{F}_t)_{t\geq 0}, \mathcal{F}, P)$ *se dice que es un* ***espacio filtrado****.*

Tomaremos $\mathcal{F}_0 = \{\phi, \Omega\}$.

Definición 5.4. *Un proceso estocástico* $X = \{X_t, t \geq 0\}$ *es un* ***proceso adaptado*** *a la filtración* $(\mathcal{F}_t)_{t\geq 0}$ *si la variable aleatoria* X_t *es una función medible de* $\mathcal{F}_t$*, para todo* $t \in \mathcal{T}$*.*

Definición 5.5. *Una sucesión* $M = (M_t)_{t\geq 0}$*, se dice que es una* ***martingala*** *(relativa a* $(\mathcal{F}_t)_{t\geq 0}$ *o* $(\mathcal{F}_t)$*-martingala) si*

$$\begin{aligned} &\textit{(i) } M \textit{ es adaptada} \\ &\textit{(ii) } \mathbb{E}(|M_t|) < \infty \\ &\textit{(iii) } \mathbb{E}(M_t|\mathcal{F}_s) = M_s, \textit{ para } s \leq t \end{aligned}$$

Una ***supermartingala*** *(relativa a* $(\mathcal{F}_t)_{t\geq 0}$*) se define similarmente excepto que (iii) es reemplazada por*

$$\mathbb{E}(M_t|\mathcal{F}_s) \leq M_s, \textit{ para } s \leq t$$

y una ***submartingala*** *se define con (iii) reemplazada por*

$$\mathbb{E}(M_t|\mathcal{F}_s) \geq M_s, \textit{ para } s \leq t$$

5.3. Proceso a tiempo discreto

Las variables en este caso son secuencias aleatorias $\{s_t(\omega), a_t(\omega)\}$ con $t \in \{0, 1, ..., T\}$, que son adaptadas a la filtración $\mathbb{F} = \{\mathcal{F}_t\}$, $t \in \{0, 1, ..., T\}$.

Suponemos que tenemos el conjunto $\{s_t, a_t\}$, $t \in \{0, 1, ..., T\}$ que denota todas las secuencias aleatorias posibles donde $s_t = s_t(\omega^t)$ y $a_t = a_t(\omega^t)$ son procesos adaptados a $\mathbb{F}$. Nuestra función valor del estado en un instante τ es:

$$V_{T-\tau}(s_\tau) = \mathbb{E}_\tau \left(\sum_{t=\tau}^{T} \gamma^{t-\tau} R(s_\tau, a_\tau) | \mathcal{F}_\tau \right)$$

Vemos que esta expresión es muy similar a la del caso determinista, la única diferencia es que ahora tenemos un promedio condicionado en τ. Previamente, para el caso determinista, definimos la función de transición de estado determinista f. Para el caso estocástico, usaremos también una función de transición:

Definición 5.6. *Una* ***función de transición de estado estocástica*** *es una función* $g : \mathcal{S} \times \mathcal{A}_s \times \mathcal{F} \longrightarrow \mathcal{S}$ *que especifica el estado del sistema en* $t + 1$ *cuando el agente elije la acción* $a \in \mathcal{A}_s$ *en el estado* s *en* t*.*

Entonces, los estados de la función valor descrita anteriormente están sujetos a la siguiente secuencia aleatoria de restricciones, dado s_0:

$$\begin{aligned} s_1 &= g(s_0, a_0) \\ s_2 &= g(s_1, a_1, \omega^2) \\ &... \\ s_{t+1} &= g(s_t, a_t, \omega^{t+1}), \text{ para } t = 2, ..., T-2 \\ &... \\ s_T &= g(s_{T-1}, a_{T-1}, \omega^T) \end{aligned}$$

donde ω es un proceso adaptado a $\mathbb{F}$, y que representa la incertidumbre que afecta a la decisión del agente.

Nuestro objetivo es encontrar la solución $\{a_t^*, s_t\}$, $t \in \{0, 1, ..., T\}$, al problema de control óptimo, por tanto será aquella secuencia que maximice la función valor

$$\max_{\{a_t\}_{t=\tau}^T} \mathbb{E}_\tau \left(\sum_{t=\tau}^{T} \gamma^{t-\tau} R(s_\tau, a_\tau) | \mathcal{F}_\tau \right)$$

Al principio del instante t, s_t y a_t son conocidos, pero el valor de s_{t+1} al final del instante t está condicionado al valor de ω^{t+1}. Los valores de este proceso aleatorio dependen pues de una variable exógena que viene dada por un proceso estocástico.

Entonces, en el instante $\tau = 0$ obtenemos

$$\begin{aligned} V_T(s_0) &= \mathbb{E}_0 \left(\sum_{t=0}^{T} \gamma^t R(s_t, a_t^*) \right) = \\ &= \max_{\{a_t\}_{t=0}^T} \mathbb{E}_0 \left(\sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right) \\ &= \max_{\{a_t\}_{t=0}^T} \mathbb{E}_0 \left(R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + ... \right) \\ &= \max_{\{a_t\}_{t=0}^T} \mathbb{E}_0 \left(R(s_0, a_0) + \gamma \sum_{t=1}^{T} \gamma^{t-1} R(s_t, a_t) \right) \\ &= \max_{a_0} \left\{ R(s_0, a_0) + \gamma \mathbb{E}_0 \left[\max_{\{a_t\}_{t=1}^T} \mathbb{E}_1 \left(\sum_{t=1}^{T} \gamma^{t-1} R(s_t, a_t) \right) \right] \right\} \end{aligned}$$

Para obtener la última igualdad hemos aplicado el principio de optimalidad. Por el hecho de que las variables son medibles respecto a $\mathcal{F}_0$, por la propiedad iterativa de la esperanza condicionada y por la definición de $V_{T-\tau}(s_\tau)$ con $\tau = 1$, vemos que:

$$V_T(s_0) = \max_{a_0} \left(R(s_0, a_0) + \gamma \mathbb{E}_0 \left(V_{T-1}(s_1) \right) \right)$$

donde s_0, a_0 y V_0 son $\mathcal{F}_0$-medible y son constantes si $\mathcal{F}_0 = (\phi, \Omega)$, y $s_1 = g(s_0, a_0, \omega^1)$ es $\mathcal{F}_1$-medible.

Si aplicamos esta misma idea para la función valor para cualquier instante $0 \leq t \leq T$, encontramos la **Ecuación de Hamilton-Jacobi-Bellman (HJB)**:

$$V_{T-t}(s_t) = \max_{a_t} \left(R(s_t, a_t) + \gamma \mathbb{E}_t \left[V_{T-t-1}(s_{t+1}) \right] \right) \tag{5.1}$$

Observamos que $\{V_{T-t}(s_t)\}$, $t \in \{0, 1, ..., T\}$, es un proceso estocástico $\mathcal{F}_t$-medible y el operador $\mathbb{E}_t(\cdot)$ es un promedio condicionado a la información disponible en el instante t (representada por $\mathcal{F}_t$).

Resolviendo la ecuación (5.1) de forma recursiva, obtendremos la solución al problema de control óptimo.

5.4. Proceso a tiempo continuo

Para desarrollar la ecuación de HJB para el caso estocástico continuo es necesario exponer previamente toda la teoría relativa a los movimientos brownianos, la integración estocástica, las ecuaciones diferenciales estocásticas, la fórmula de Itô, etc. Es por esto que en esta sección solo nos limitaremos a comentar de forma general cómo obtenemos la ecuación de HJB sin justificarlo de forma muy precisa.

Nuestro objetivo está en encontrar funciones $(s^*(t), a^*(t))$ que resuelvan el siguiente problema.

$$\max_a \mathbb{E} \left(\int_{t_0}^{T} R(t, s(t), a(t)) dt \right)$$

sujeto a la ecuación diferencial estocástica

$$ds(t) = g(t, s(t), a(t))dt + \sigma(t, s(t), a(t))dB(t)$$

dada la distribución inicial de la variable de estado $s(0, \omega) = s_0(\omega)$. Llamamos $g(\cdot)$ a la función de transición de estado estocástico. La función de recompensa $R(\cdot)$ y la función $\sigma(\cdot)$ son funciones $\mathcal{F}_t$-adaptadas.

La función valor del estado para el instante inicial es

$$\mathcal{V}(t_0, s_0) = \mathbb{E} \left(\int_{t_0}^{T} R(t, s^*, a^*) dt \right)$$

Lema 5.7. *Sea $\mathcal{V} \in \mathcal{C}^2(\mathcal{T}, \mathbb{R})$. La función valor asociada al camino óptimo $\{(s^*(t), a^*(t)) : t_0 \leq t \leq T\}$ verifica la **Ecuación de Hamilton-Jacobi-Bellman (HJB)**:*

$$-\mathcal{V}_t(t, s) = \max_a \left(R(t, s, a) + g(t, s, a)\mathcal{V}_s(t, s) + \frac{1}{2}\sigma^2(t, s, a)\mathcal{V}_{ss}(t, s) \right) \tag{5.2}$$

donde $\mathcal{V}_t(t, s) := \frac{\partial \mathcal{V}(t,s)}{\partial t}$, $\mathcal{V}_s(t, s) := \frac{\partial \mathcal{V}(t,s)}{\partial s}$ y $\mathcal{V}_{ss}(t, s) := \frac{\partial^2 \mathcal{V}(t,s)}{\partial s^2}$.

Demostración. La demostración es similar a la del caso determinista continuo. Consideramos la función valor

$$\mathcal{V}(t, s) = \max_a \mathbb{E} \left(\int_t^T R(u, s, a) du \middle| \mathcal{F}_t \right)$$

Queremos encontrar una ecuación diferencial parcial para $\mathcal{V}(t,s)$. Suponemos en todo momento que $R(\cdot)$ es continua como función total de t, con $s = s(t)$ y $a = a(t)$. Para todo $h > 0$ y usando el principio de optimalidad podemos escribir

$$\begin{aligned}\mathcal{V}(t,s) &= \max_a \mathbb{E}\left[\left(\int_t^{t+h} R(u,s,a)du + \int_{t+h}^{T} R(u,s,a)du\right)\middle|\mathcal{F}_t\right] \\ &= \max_a \mathbb{E}\left[\left(\int_t^{t+h} R(u,s,a)du + \mathbb{E}\left(\int_{t+h}^{T} R(u,s,a)du\middle|\mathcal{F}_{t+h}\right)\right)\middle|\mathcal{F}_t\right] \\ &= \max_a \mathbb{E}\left[\int_t^{t+h} R(u,s,a)du + \mathcal{V}(t+h, s_{t+h})\middle|\mathcal{F}_t\right]\end{aligned}$$

Asumiendo que $\mathcal{V}$ es una función continua y diferenciable de segundo orden, aplicando la formula de Itô obtenemos

$$\begin{aligned}\mathcal{V}(t+h, s_{t+h}) = \mathcal{V}(t,s) &+ \int_t^{t+h}\left(\frac{\partial \mathcal{V}}{\partial u} + \frac{\partial \mathcal{V}}{\partial s}g(u,s,a) + \frac{1}{2}\frac{\partial^2 \mathcal{V}}{\partial s^2}\sigma^2(u,s,a)\right)du \\ &+ \int_t^{t+h}\frac{\partial \mathcal{V}}{\partial s}\sigma(u,s,a)dB_u\end{aligned}$$

Como consecuencia, si $\int_{t_0}^{T}\mathbb{E}\left(\mathcal{V}_s\sigma_t\right)^2 dt < \infty$,

$$\mathbb{E}(\mathcal{V}(t+h, s_{t+h})|\mathcal{F}_t) = \mathcal{V}(t,s) + \mathbb{E}\left(\int_t^{t+h}\left(\mathcal{V}_u + \mathcal{V}_s g(u,s,a) + \frac{1}{2}\mathcal{V}_{ss}\sigma^2(u,s,a)\right)du\middle|\mathcal{F}_t\right)$$

y por tanto, para todo $h > 0$

$$0 = \mathbb{E}\left(\int_t^{t+h}\max_a\left(R(u,s,a) + \mathcal{V}_u + \mathcal{V}_s g(u,s,a) + \frac{1}{2}\mathcal{V}_{ss}\sigma^2(u,s,a)\right)du\middle|\mathcal{F}_t\right)$$

Como esto es cierto para todo $h > 0$ y $t_0 \le t \le T$, obtenemos que

$$\mathcal{V}_t + \max_a\left(R(t,s,a) + \mathcal{V}_s g(t,s,a) + \frac{1}{2}\mathcal{V}_{ss}\sigma^2(t,s,a)\right) = 0$$

□

6. Aplicación a Reinforcement Learning

El aprendizaje por refuerzo (conocido como *Reinforcement Learning*) es una de las categorías principales de *Machine Learning*. El aprendizaje automático o *Machine Learning* es una disciplina del campo de la inteligencia artificial que, a través de algoritmos, dota a los ordenadores de la capacidad de identificar patrones en datos masivos y elaborar predicciones. Este aprendizaje permite a los ordenadores realizar tareas específicas de forma autónoma, es decir, sin necesidad de ser programados. Las técnicas de aprendizaje automático son actualmente una parte fundamental del *Big Data*. Algunos sistemas de *Machine Learning* intentan eliminar toda necesidad de intuición o conocimiento experto de los procesos de análisis de datos, mientras otros tratan de establecer un marco de colaboración entre el experto y el ordenador.

El aprendizaje automático tiene una amplia gama de aplicaciones, como por ejemplo en motores de búsqueda (sistemas informáticos que buscan archivos almacenados en servidores web), diagnósticos médicos, detección de fraude en el uso de tarjetas de crédito, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, juegos, robótica...

Existen distintos algoritmos de *Machine Learning*, que se dividen en tres categorías, una de las cuales es en la que nos centraremos (*Reinforcement Learning*).

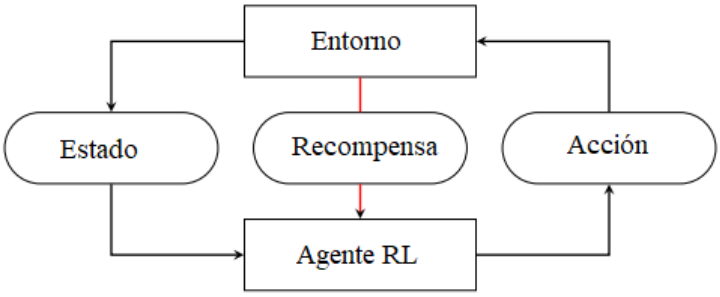
- **Aprendizaje supervisado**: Estos algoritmos cuentan con un aprendizaje previo basado en un sistema de etiquetas asociadas a unos datos que les permiten tomar decisiones o hacer predicciones. El algoritmo produce una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Un ejemplo podría ser un detector de *spam*, que etiqueta un e-mail como *spam* o no dependiendo de los patrones que ha aprendido del histórico de correos (como el remitente, relación texto e imágenes, palabras clave en el asunto,etc).
- **Aprendizaje no supervisado**: Estos algoritmos no cuentan con un conocimiento previo. Se enfrentan al caos de datos con el objetivo de encontrar patrones que permitan organizarlos de alguna manera. Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan solo por entradas al sistema, pero sin información sobre las categorías de esos ejemplos. Por ejemplo, en el campo del *marketing* se utilizan para extraer patrones de datos masivos provenientes de las redes sociales y crear campañas de publicidad altamente segmentadas.
- **Aprendizaje por refuerzo o *Reinforcement Learning***: Su objetivo es que un algoritmo aprenda a partir de la propia experiencia, es decir, que sea capaz de tomar la mejor decisión ante diferentes situaciones de acuerdo a un proceso de prueba y error en el que se recompensan las decisiones correctas. Por ejemplo, actualmente se usa para posibilitar el reconocimiento facial, hacer diagnósticos médicos o clasificar secuencias de ADN.

6.1. Qué es *Reinforcement Learning*

Reinforcement Learning (RF) es el área del aprendizaje automático cuya ocupación es determinar qué acciones debe escoger un agente de software en un entorno dado con el fin de maximizar alguna noción de recompensa o premio acumulado. Una gran parte de los algoritmos usados en el aprendizaje por refuerzo se fundamentan en la programación dinámica. Se considera que el aprendizaje por refuerzo es una extensión de la programación dinámica que proporciona soluciones sin la necesidad de conocer el modelo de comportamiento del sistema, por tanto, entra dentro de los procesos estocásticos.

Los principales elementos del problema que se quiere resolver en RF y su interacción son los que habíamos descrito para la programación dinámica. Como vemos en la Figura 2, el agente interactúa con el entorno mediante tres elementos: el estado del entorno ($s_t \in \mathcal{S}$), la acción que permite al agente influenciar el estado del entorno ($a_t \in \mathcal{A}$) y una función de recompensa (R_t), la cual proporciona al agente información sobre la calidad de la acción que acaba de realizar en el estado actual. En cada instante temporal, el agente recibe una medida del estado y realiza una acción. Como consecuencia, se produce una transición del

Figura 2: Situación estándar en RL



entorno a un nuevo estado. Además se genera una recompensa que evalúa la calidad de dicha transición. Entonces el agente recibe el nuevo estado y el ciclo completo se repite. El agente selecciona la acción realizada en cada estado de acuerdo a una estrategia, que es una función de estados a acciones. El objetivo del agente es aprender una estrategia que maximice la cantidad total de recompensa recibida, es decir, la recompensa acumulada a largo plazo.

El proceso de aprendizaje se basa en los datos que adquiere el agente mediante la interacción con el entorno. Cada vez que el agente realiza una acción, la información que recibe por parte del entorno, es decir, del estado siguiente y la recompensa, es utilizada para modificar la estrategia. Normalmente se usa el término *experiencia* para los datos adquiridos por el agente. Durante las primera etapas del aprendizaje, las acciones realizadas se seleccionan de forma aleatoria y, conforme aumenta la experiencia, la estrategia mejora de forma que las acciones proporcionan al agente una mayor cantidad de recompensa a largo plazo.

Al estar el RL asentado sobre una sólida base teórica y como muchos de sus algoritmos están basados en los principios de funcionamiento de la programación dinámica, este ha permitido el desarrollo de análisis teóricos que demuestran bajo qué condiciones, y a qué velocidad, convergen los algoritmos hacia estrategias óptimas. Aunque se pueden resolver un gran número de problemas mediante RL, su uso está limitado principalmente por el crecimiento exponencial de los requisitos computacionales y de su almacenamiento cuando el número de variables de estado y de acción se incrementa. Además, el algoritmo de aprendizaje por refuerzo debe seleccionarse cuidadosamente ya que sino es probable que el proceso de aprendizaje no converja hacia una estrategia óptima o que la convergencia no esté garantizada.

Antes de explicar algunos de los principales algoritmos de RL, presentaremos varios ejemplos.

6.2. Ejemplos

Referencia: *Reinforcement Learning: An introduction* (2018). Richard Sutton, Andrew G. Barto.

Una buena forma de entender el aprendizaje por refuerzo es considerando algunos de los ejemplos y posibles aplicaciones que han guiado su desarrollo.

- Un jugador de ajedrez realiza un movimiento. La elección que toma es por un lado planificada (anticipando posibles respuestas y contrarrespuestas) y por otro lado por juicios inmediatos e intuitivos (desando ocupar ciertas posiciones o realizar ciertos movimientos).
- Un controlador ajusta los parámetros de operación de una refinería de petróleo en tiempo real. El controlador optimiza el rendimiento, el coste, la calidad y la compensación sobre la base de costos marginales especificados sin ceñirse estrictamente a los puntos de referencia originalmente sugeridos por los ingenieros.
- Una cría de gacela lucha por ponerse en pie minutos después de nacer. Media hora después ya corre a 30 km/h.
- Un robot móvil decide si debe entrar en una nueva habitación en busca de más basura para recolectar o comenzar a tratar de encontrar el camino de regreso a su estación para recargar la batería. Toma su decisión basándose en su nivel de carga de la batería actual y cómo de rápido y fácil ha sido encontrar el cargador en el pasado.
- Una persona prepara su desayuno. Examinada de cerca, incluso esta actividad aparentemente mundana revela una compleja red de comportamientos, como caminar hacia el armario, abrirlo, seleccionar una caja de cereales, alcanzarla, cogerla... Se requieren también otras secuencias de comportamiento complejas e interactivas para obtener un bol, una cuchara, la leche, etc. Cada paso implica una serie de movimientos oculares para obtener información y para guiar al alcance de cada objeto. Continuamente se hacen juicios rápidos acerca de cómo llevar cada objeto. Cada paso está guiado por objetivos, como coger la cuchara o llegar a la nevera, y está al servicio de otros objetivos, como usar la cuchara para comer una vez preparada la comida y obtener alimento. Tanto si esta persona es consciente o no de ello, está accediendo a información sobre el estado de su cuerpo que determina sus necesidades nutricionales, nivel de hambre y preferencias.

Todos estos ejemplos comparten características y todos implican la interacción entre un agente activo que toma las decisiones y su entorno, dentro del cual el agente busca lograr un objetivo a pesar de la incertidumbre que hay. Vemos que las acciones del agente afectan el estado futuro del entorno (por ejemplo, la próxima posición en el ajedrez, el nivel de depósitos de la refinería, el futuro nivel de carga del robot), afectando así las opciones y oportunidades disponibles para el agente en momentos posteriores. La decisión correcta requiere tener en cuenta los efectos o posibles consecuencias que tengan esas acciones y por tanto, puede requerir previsión o planificación. Al mismo tiempo, en todos estos ejemplos los efectos de las acciones no siempre se pueden predecir completamente. Por tanto, el agente debe monitorear su entorno con frecuencia y reaccionar apropiadamente.

Todos estos ejemplos implican metas que son explícitas en el sentido de que el agente puede juzgar el progreso hacia su meta basado en lo que puede observar directamente. Por ejemplo, el jugador de ajedrez sabe si gana o no; el controlador de la refinería sabe

cuánto petróleo se está produciendo; el robot sabe cuándo se acaban las pilas, etc.

Vemos en estos ejemplos que tanto el agente como su entorno pueden tomar muchas formas distintas. Además, vemos en los ejemplos que el agente puede usar su experiencia para mejorar su ejecución con el tiempo. El jugador de ajedrez mejora la intuición que usa para evaluar las posiciones y mejorando así su juego; la cría de gacela mejora la eficiencia con la que corre; la persona agiliza la preparación del desayuno. El conocimiento que el agente aporta a la tarea desde el principio, ya sea por la experiencia previa o a través de su evolución, influye en lo que es útil o fácil de aprender, pero la interacción con el entorno es fundamental para ajustar el comportamiento y explotar las características específicas de la tarea.

A continuación, presentaremos algunos de los principales algoritmos de RL, centrándonos en aquellos basados en el aprendizaje *temporal difference*. El objetivo sigue siendo exactamente el mismo que en programación dinámica: encontrar una estrategia que maximice la recompensa obtenida por el agente. La principal diferencia es que los algoritmos de RL están basados en la experiencia en lugar del modelo. Por otra parte, dado que la estrategia se obtiene a partir de las muestras que adquiere el agente, es necesario que el proceso de muestreo cumpla ciertas propiedades estadísticas para asegurar que la estrategia sea óptima.

6.3. *Temporal difference*

Temporal difference (TD) hace referencia a una familia de métodos para estimar o predecir la función valor V de una estrategia fija, y también puede ser extendido a las función valor del estado-acción Q. Cada vez que el agente realiza una acción, el algoritmo TD usa la recompensa generada y la estimación actual de V para realizar una nueva estimación de acuerdo a la siguiente expresión

$$V_{t+1}(s_t) = V_t(s_t) + \alpha_t[R_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)] \tag{6.1}$$

donde $\alpha_t \in [0, 1]$ es la secuencia de tasas de aprendizaje que determina la cantidad con la que se actualiza el valor del estado s_t. El término

$$R_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)$$

se conoce como **diferencia temporal**, y es la diferencia entre la nueva estimación de la función V, $R_{t+1} + \gamma V_t(s_{t+1})$, y la estimación en el instante anterior, $V_t(s_t)$. La ecuación utilizada para actualizar V puede expresarse como:

$$\text{estimación}_{\text{nueva}} \leftarrow \text{estimación}_{\text{anterior}} + \alpha[\text{objetivo} - \text{estimación}_{\text{anterior}}]$$

El algoritmo *temporal difference* se puede resumir de la siguiente forma:

1. Se da como entrada: el factor de descuento γ, la secuencia de tasas de aprendizaje α_t y la estrategia π.
2. Se inicia la estimación de la función V arbitrariamente (por ejemplo, $V_0 = 0$).
3. Para cada estado s_t (empezando por s_0), se efectúa la acción $a_t = \pi(s_t)$.

4. Al aplicar a_t, se observa el siguiente estado s_{t+1} y la recompensa R_{t+1}.
5. Se aplica $V_{t+1}(s_t) = V_t(s_t) + \alpha_t[R_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)]$.
6. Se repite desde (3) hasta cumplir con las condiciones de convergencia.
7. Finalmente se devuelve V_π.

A continuación procedemos a presentar dos algoritmos muy conocidos en RL, basados en TD:

- **SARSA**: Es un método *on-policy*, que significa que la acción de la función valor es estimada por la estrategia actual y por el par estado-acción.
- ***Q-learning***: Es un método *off-policy*, que significa que la acción óptima de la función valor es estimada independientemente de la estrategia actual.

6.3.1. SARSA

Dado que TD aprende a partir de tuplas de muestras de la forma $(s_t, a_t, R_{t+1}, s_{t+1}, a_{t+1})$, el nombre SARSA viene de unir las iniciales de cada nombre en la tupla de datos empleada por el algoritmo: estado (*state*), acción (*action*), recompensa (*reward*), (siguiente) estado y (siguiente) acción. Este algoritmo se basa en actualizar la función valor estado-acción Q de la siguiente manera:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \left[R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right] \tag{6.2}$$

donde $\alpha_t \in [0, 1]$ es la secuencia de tasas de aprendizaje. Esta actualización se hace en cada uno de los instantes, y las acciones se elijen mediante una estrategia *greedy*, es decir, una estrategia que maximice la función valor.

Por tanto, el algoritmo SARSA es el siguiente.
Entrada: El factor de descuento γ y la secuencia de tasas de aprendizaje α_t.
Salida: La estrategia óptima π^* aproximada.

1. Se inicia $Q(s, a)$ arbitrariamente para todo $(s, a) \in \mathcal{S} \times \mathcal{A}$.
2. Se inicia $Q(s, \cdot) = 0$ para todos los estados finales.
3. Se repite para cada instante t, hasta llegar al último instante:
    1) Se inicia s.
    2) Se escoge la acción a usando una estrategia *greedy* a partir de la Q actual.
    3) Se repite para cada paso en el instante t, hasta llegar al último estado:
        1) Se toma la acción a y se observa la recompensa R y el estado siguiente s'.
        2) Se escoge la acción a' usando una estrategia *greedy* a partir de la Q actual.
        3) Actualizamos el valor de Q mediante la expresión 6.2, es decir, $Q(s, a) \leftarrow Q(s, a) + \alpha[R + \gamma Q(s', a') - Q(s, a)]$.
        4) $s \leftarrow s'$ y $a \leftarrow a'$.
4. Para todo $s \in \mathcal{S}$ se hace $\pi(s) \leftarrow arg\,\text{máx}_{a \in \mathcal{A}}\, Q(s, a)$.
5. Se devuelve π.

6.3.2. *Q-learning*

Q-learning es una extensión del algoritmo TD al problema general de aprender estrategias óptimas. A partir de una estimación arbitraria de la función Q, el algoritmo emplea la tupla $(s_t, a_t, s_{t+1}, R_{t+1})$ para actualizar la estimación de Q mediante la regla:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \left[R_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \tag{6.3}$$

siendo $\alpha_t \in [0, 1]$ la secuencia de tasas de aprendizaje.

El principio de funcionamiento de *Q-learning* consiste en estimar de forma iterativa la función Q óptima. Este algoritmo es de tipo *off-policy* ya que converge hacia una estrategia óptima independiente de la estrategia que emplee el agente para interactuar con el entorno. Para converger, *Q-learning* requiere que todos los pares estado-acción de la función Q sean actualizados indefinidamente, es decir, una estrategia exploratoria.

El algoritmo *Q-learning* es el siguiente.
Entrada: El factor de descuento γ y la secuencia de tasas de aprendizaje α_t.
Salida: La estrategia óptima π^* aproximada.

1. Se inicia $Q(s, a)$ arbitrariamente para todo $(s, a) \in \mathcal{S} \times \mathcal{A}$.
2. Se inicia $Q(s, \cdot) = 0$ para todos los estados finales.
3. Se repite para cada instante t, hasta llegar al último instante:
    1) Se inicia s.
    2) Se repite para cada paso en el instante t, hasta llegar al último estado:
        1) Se escoge la acción a usando una estrategia *greedy* a partir de la Q actual.
        2) Se toma la acción a y se observa la recompensa R y el estado siguiente s'.
        3) Actualizamos el valor de Q mediante la expresión 6.3, es decir, $Q(s, a) \leftarrow Q(s, a) + \alpha[R + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s, a)]$.
        4) $s \leftarrow s'$.
4. Para todo $s \in \mathcal{S}$ se hace $\pi(s) \leftarrow arg\max_{a \in \mathcal{A}} Q(s, a)$.
5. Se devuelve π.

6.4. Limitaciones de RF

Hemos visto que el aprendizaje por refuerzo es muy útil y tiene muchas aplicaciones, por ejemplo en el campo de las finanzas. Pero para finalizar, vemos que no está exento de limitaciones o desafíos, algunos de ellos son:

- Especificación correcta del modelo.

Esto implica que la representación del estado, la elección de las acciones y el diseño de la recompensa son específicos de cada problema que queremos tratar, tanto en términos de especificación como en dificultad.

- Obtención de datos suficientes para “aprender“.

Muchas veces se dispondrán de pocos datos y será difícil que el algoritmo aprenda mediante la experiencia. Para enfrentar este problema, se desarrolla primero un modelo estocástico del entorno con relativamente pocos parámetros, ajustando esos pocos parámetros a datos del historial y luego entrenando un sistema RL en tantas simulaciones del modelo estocástico como sea necesario. Por tanto, se utiliza un modelo estocástico para imputar los datos faltantes del entorno. Mientras esto puede servir como punto de partida, el aprendizaje se vuelve dependiente de las suposiciones y especificaciones del modelo estocástico elegido.

- Entorno no-Markoviano.

En el aprendizaje por refuerzo se asume que el entorno es Markoviano, y muchas veces no es así. El modelo Markoviano describe una secuencia de estados posibles en la que la probabilidad de cada uno depende únicamente del estado alcanzado anteriormente.

- Conjunto de estados y acciones de gran dimensión.

Muchos problemas prácticos del mundo real tienen estados y espacios de acción muy grandes y continuos. Esto puede presentar serios problemas para los algoritmos usados en RL.

- Recompensa no especificada.

El aprendizaje por refuerzo se basa en buscar estrategias mediante la optimización de una función de recompensa global. Pero la mayoría de sistemas tienen una función de recompensa multidimensional (por ejemplo, reducir el tiempo de ejecución pero también el consumo de energia) que deben minimizarse. En muchos casos no se tiene una idea clara de lo que se quiere optimizar. Es por esto que gran parte del trabajo a la hora de implementar RL consiste en formular correctamente la función de recompensa.

7. Conclusiones

En este trabajo nos hemos adentrado en el método de optimización llamado programación dinámica inventado por Richard Bellman, un método para resolver problemas de tipo secuencial con el propósito de encontrar la mejor solución posible entre todas las soluciones viables.

Hemos definido el entorno en el que se mueve el agente a la hora de enfrentarse a un problema de toma decisiones, con conceptos como los estados, las acciones que llevan al agente a estar en un estado u otro, la recompensa que recibe por realizar una acción determinada y las estrategias que sirven para seleccionar la mejor acción a tomar en cada estado. Y por último hemos introducido la función valor, una función que determina lo que es bueno para el agente a largo plazo y es útil para encontrar y comparar estrategias.

A continuación, hemos introducido también los procesos de decisión Markovianos (MDP), modelos para la toma de decisiones en el tipo de problemas que hemos tratado en el trabajo. El objetivo principal al tratar con MDP en programación dinámica ha sido encontrar la función valor para una estrategia dada, y encontrar la estrategia que nos lleve a la función valor óptima. La estrategia óptima es aquella que maximiza la recompensa que espera recibir el agente a largo plazo, es decir, indica al agente cuáles son las mejores acciones a tomar. Para ello hemos introducido el Principio de Optimalidad de Bellman, que es fundamental para resolver estos problemas usando cálculos recursivos, y que mediante este principio, hemos podido desarrollar las ecuaciones de Bellman y posteriormente las ecuaciones de Hamilton-Jacobi-Bellman (HJB).

Para desarrollar las ecuaciones de HJB, hemos hecho distinción de diferentes tipos de modelos: enfoque determinista (cuando el estado siguiente se encuentra determinado por completo respecto al estado y decisión actual) y enfoque estocástico (cuando el estado de la siguiente etapa queda determinado mediante una distribución de probabilidad). Para cada uno de estos enfoques hemos hecho también distinción según si el tiempo evolucionaba de forma discreta o continua. Para cada uno de estos casos hemos encontrado la ecuación de HJB. Para el proceso determinista hemos resuelto un ejemplo con tal de ver con más claridad cómo se aplica la programación dinámica.

Finalmente hemos hablado del aprendizaje por refuerzo (*reinforcement learning*, RL) como una extensión de la programación dinámica. Ésta es un área de la inteligencia artificial centrada en determinar qué acciones debe escoger un agente de software en un entorno dado con el fin de encontrar la máxima recompensa. Para acabar, hemos presentado dos de los principales algoritmos de RL: SARSA y *Q-learning*.

Referencias

[1] Bacchus, Fahiem; Boutilier, Craig; Gorve, Adam. *Structured Solution Methods for Non-Markovian Decision Processes*. Proc. 14th National Conf. on AI (AAAI-97), Providence, August, 1997.

[2] Bellman, Richard. *Dynamic Programming*. 2ª ed. Introduction by Dreyfus, Stuart. United Kingdom: Princeton University Press, 1957. 339 p. ISBN 978-0-691-14668-3.

[3] Brito, Paulo. *Introduction to Dynamic Programming Applied to Economics*. Lisboa: Universidade Técnica de Lisboa, Instituto Superior de Economia y Gestao, 2007. Disponible a: `https://www.fep.up.pt/docentes/joao/material/aea/notas_pbrito_2007.pdf`

[4] Corcuera, José M. *Apuntes del curso de Finanzas Cuantitativas*. Universidad de Barcelona.

[5] Díaz Iza, Henry P. *Programación dinámica y aprendizaje por refuerzo*, Trabajo Fin de Máster. Director: Antonio Sala Piqueras. Valencia: Universidad Politécnica de Valencia, Departamento de ingeniería de sistemas y automática, 2015.

[6] Dulac-Arnold, Gabriel; Mankowitz, Daniel; Hester, Todd. *Challenges of Real-World Reinforcement Learning*. Cornell University, april 2019. Data provided by arXiv:1904.12901.

[7] Fernández-Villaverde, Jesús; Nuño, Galo. *Dynamic programming in continuous time*. Pennsylvania: University of Pensylvannia, Department of Economics, octubre 2021, [consulta: 19 diciembre 2021]. Disponible a: `https://www.sas.upenn.edu/ jesusfv/Continuous_Time_1.pdf`

[8] García-Ocaña Hernández, Daniel M. *Estudio y desarrollo de algoritmos de aprendizaje por refuerzo a partir de la teoría de optimización dual*, Tesis fin de Máster. Director: D. Santiago Zazo Bello. Madrid: Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación, 2017.

[9] Kirk, Donald E. *Optimal Control Theory: An Introduction*. New York: Dover Publications, Inc., 1970. 443 p. ISBN 0-486-43484-2.

[10] Kolm, Petter N.; Ritter, Gordon. *Modern Perspectives on Reinforcement Learning in Finance*. The Journal of Machine Learning in Finance, Vol. 1, No. 1, 2020. Diponible a SSRN: `https://ssrn.com/abstract=3449401` o `http://dx.doi.org/10.2139/ssrn.3449401`

[11] Maurette, Manuel; Ojea, Ignacio. *Programación dinámica*. Buenos Aires: Facultad de ciencias exactas y naturales, Universidad de Buenos Aires, junio 2006, [consulta: 23 octubre de 2021]. Disponible a: `http://cms.dm.uba.ar/materias/1ercuat2009/optimizacion/Maurette_Ojea.pdf`

[12] Pedraza Pina, Jorge C. *Biografia Richard Bellman*. 2015 [consulta: 21 diciembre de 2021]. Disponible a: `https://jorgeperaza0412.wordpress.com/2015/08/20/biografia-richard-bellman/`

[13] Puterman, Martin L. *Markov Decision Processes: Discrete Stochastic. Dynamic Programming*. 2ª ed. New Jersey: John Wiley & Sons, Inc., 1994. 643 p. ISBN 0-471-72782-2.

[14] Rincón, Luis. *Construyendo la integral estocástica de Itô*. México: Universidad Nacional Autónoma de México (UNAM), 2004 [consulta: 2 enero de 2022]. Disponible a: `https://lya.fciencias.unam.mx/lars/pub/ede.pdf`

[15] Stokey, Nancy L., Lucas, Robert E., Prescott, Edward C. *Recursive Methods in Economic Dynamics*. 5ª ed. United States: President and Fellows of Harvard College, 1999. 608 p. ISBN 0-674-75096-9.

[16] Sutton, Richard S.; Barto, Andrew G. *Reinforcement Learning: An introduction*. 2ª ed. United States: Westchester Publishing Services, 2018. 519 p. ISBN 9780262039246.

[17] Torres, Jordi. *Funciones de valor y la ecuación de Bellman*. Barcelona: UPC Barcelona Tech & Barcelona Supercomputing Center, abril 2021 [consulta: 15 octubre de 2021]. Disponible a: `https://medium.com/aprendizaje-por-refuerzo/3-funciones-de-valor-y-la-ecuación-de-bellman-7b0ebfac2be1`

[18] Wilmott, Paul. *Machine Learning: An applied mathematics introduction*. 1ª ed. Panda Ohana Publishing, 2019. 223 p. ISBN 978-1-9160816-0-4.