clave unica _____

Bandidos de Markov de Dos Brazos

Una **Guía Paso a Paso** para Modelar, Problemas de Decisión Básicos y Variaciones

Nota: Incluimos la notación como si las probabilidades de transición dependieran de A_t (la acción tomada por el agente). Sin embargo, para simplificar los ejercicios y todo lo demás, se puede asumir que la acción no afecta las probabilidades de transición y que son independientes de las acciones.

EL primer objetivo es resolver la **seccion 6** de ejercicios para plantear un problema de decicion para una verion modificada, o resolver el ejercicio computacional que biene descrito en el resto de las secciones.

1. Introducción y Contexto

Los problemas de bandido de dos brazos son un escenario clásico en **aprendizaje por refuerzo** y **teoría de la decisión**, donde un agente elige uno de dos "brazos" en cada paso de tiempo para recibir una recompensa. Mientras que la versión **i.i.d. (independiente e idénticamente distribuida)** del problema asume que la distribución de recompensas de cada brazo es fija en el tiempo, una variante **Markov** permite que la probabilidad (o nivel) de recompensa de cada brazo **evolucione** con el tiempo. Esta evolución está gobernada por un proceso de Markov, que puede estar correlacionado entre los dos brazos.

1.1 Motivación

¿Por qué pasar de bandidos i.i.d. a bandidos **Markov**?

1. **Recompensas Dependientes del Estado**
   En muchos casos reales, la recompensa que produce un bandido puede depender de un **estado** interno que cambia con el tiempo. Por ejemplo, una máquina podría "desgastarse" si se elige repetidamente, lo cual reduce su pago si permanece en un estado "malo".

2. **Observabilidad Parcial**
   A menudo, el agente **no** ve el estado real. Solo ve la recompensa del brazo elegido (o alguna señal parcial). Por lo tanto, debe mantener una **creencia** acerca de los estados de Markov subyacentes.

3. **Bono de Exploración**
   Para evitar la explotación puramente miope (siempre elegir el brazo que aparenta ser mejor), podemos añadir un **bono de exploración** β,B_t. Este término extra recompensa al agente por acciones que potencialmente descubren nueva información sobre los estados o transiciones de los brazos.

PROF

Al tratar con **bandidos de Markov de dos brazos**, se mantienen todos los intercambios clásicos — exploración vs. explotación, conocimiento parcial o total de las transiciones, horizonte finito o infinito— pero con la complejidad adicional que aporta la evolución de estados de Markov.

2. Del Escenario Real al Problema de Decisión

Traduzcamos un escenario práctico de bandido a un problema de decisión formal—concretamente un Proceso de Decisión de Markov (MDP) o un MDP Parcialmente Observable (POMDP) cuando el estado subyacente no es observado directamente.

2.1 Formulación MDP/POMDP Paso a Paso

A continuación, se muestra la estructura general que utilizaremos en este documento:

1. **Estados**
   Cada brazo tiene un estado $X_t^{(1)}$ y $X_t^{(2)}$ en el tiempo t. Podemos representar el estado global (oculto) como
   [
   X_t ;=; \bigl(X_t^{(1)},; X_t^{(2)}\bigr).
   ]
   Importante: El agente **no** observa X_t. Conoce (o aprende) cómo evoluciona X_t, pero nunca lo ve directamente.

2. **Acciones**
   En cada instante t, el agente elige qué brazo jalar:
   [
   A_t ;\in; {,1,,2}.
   ]

3. **Transición**
   El estado de los bandits evoluciona después de cada tirón. Esta evolución puede ser:

   - **Independiente**: El estado de cada brazo evoluciona según su propia matriz de transición.
   - **Dependiente (conjunta)**: Los dos brazos evolucionan de acuerdo con una única distribución de transición *conjunta* $P(X_{t+1} \mid X_t, A_t)$.

PROF

4. **Observaciones**
   Debido a que el agente **nunca** ve el estado subyacente directamente, solo recibe observaciones parciales. El ejemplo más simple es que ve la **recompensa** R_t del brazo elegido. En algunas variaciones, puede ver señales adicionales con ruido correlacionadas con X_{t+1}.

5. **Recompensas**
   Generalmente, cada vez que el agente elige el brazo i, observa una recompensa R_t. En la forma más simple, R_t podría ser **Bernoulli** (0 o 1) con una probabilidad dependiente del estado actual (oculto) del brazo elegido. Más generalmente, podemos tener recompensas de valor real.

6. **Bono de Exploración**
   Se puede añadir un término adicional β , B_t para fomentar la exploración. La función B_t podría ser pequeña si tenemos mucha información sobre estados futuros, o grande si hay incertidumbre. β es un parámetro definido por el usuario que pondera la importancia de la exploración.

7. **Política**
   Una política, denotada Pol, es una asignación desde la información actual del agente (p. ej., una **creencia** sobre los estados o simplemente el historial de recompensas) a una acción en ${1,2}$. Formalmente,
   [
   A_t ;=; \text{Pol}(I_t),
   ]
   donde I_t es la información del agente en el instante t.

8. **Objetivo**
   Un objetivo común es **maximizar** la recompensa acumulada esperada más el bono de exploración:
   [
   \max_{\text{Pol}} ;\mathbb{E}^{\text{Pol}}!\Bigl[\sum_{t=1}^{T} \bigl(R_t + \beta,B_t\bigr)\Bigr],
   ]
   en un horizonte **finito** T. Hay variantes con T aleatorio o infinito.

2.2 Generación Probabilística en Segundo Plano

En realidad cada bandido o banda tiene una probabilidad de otrogar un premio o no (ganar o perder) en cada turno, su probabilidad esta modelada como una cadena de markov donde la probabilidad de ganar depende de la probabilidad anterior. El resto de esta seccion explica como se modela a nivel computacional, pero es una nota tecnica en estricto sentido.

2.2.1 Nota tecnica

En el trasfondo, los parámetros de cada problema (matrices de transición, probabilidades de recompensa) pueden **generarse aleatoriamente** mediante un **método basado en Dirichlet**. Concretamente:

- **Matrices de Transición**
  Para cada fila de la matriz de transición de Markov (ya sea para cada brazo de forma independiente o para el estado conjunto), se dibuja un vector aleatorio de $\text{Dirichlet}(1,1,\dots,1)$. Esto asegura que cada fila sea una distribución de probabilidad válida que sume 1, tratando todas las posibilidades de siguiente estado de manera uniforme en promedio.

PROF

- **Recompensas**
  Si la recompensa es Bernoulli, su probabilidad de éxito puede dibujarse desde una distribución **Beta** (equivalente a $\text{Dirichlet}(1,1)$). Si hay múltiples niveles de recompensa o recompensas más complejas, se aplica un método similar de muestreo con Dirichlet.

---

3. Problemas de Decisión Básicos (Casos Base, *sin Heurísticas*) y Derivación de Todas las Variaciones

Esta sección define un conjunto mínimo de **problemas de decisión básicos**—cada uno con **detalle riguroso y formal**—y luego muestra cómo **cualquier** otra variación (independientes vs. dependientes, transiciones conocidas vs. desconocidas, horizonte fijo o infinito, etc.) puede derivarse o mapearse a estos casos base.

> **Nota**: En **todos** los problemas básicos, el agente nunca observa directamente X_t. Lo que difiere es si la matriz de transición es conocida/desconocida, si el horizonte es fijo/infinito, etc. Las observaciones parciales del agente suelen provenir de la recompensa del brazo elegido.

3.1 Convenciones de Notación

- $X_t^{(1)}, X_t^{(2)}$ = estados ocultos de los brazos 1 y 2 en el tiempo t.
- $X_t = (X_t^{(1)},,X_t^{(2)})$ = el estado conjunto oculto.
- $A_t \in {1,2}$ = acción (elegir qué brazo jalar).
- R_t = recompensa observada en el tiempo t.
- β = coeficiente del bono de exploración.
- B_t = función de bono de exploración en el tiempo t.
- Pol = política; asigna la **información** del agente (p. ej., creencia o historial de recompensas) a ${1,2}$.
- T = horizonte (posiblemente fijo, aleatorio o infinito).

3.2 Los Cuatro Problemas Básicos

Problema Básico 1

Independiente, Transiciones Conocidas, T Fijo, (Estado Oculto) + Bono de Exploración

1. **Espacio de Estados**
   [
   \mathcal{X}^{(i)} \text{ es el conjunto finito de posibles estados para el brazo } i,\quad i\in{1,2}.
   ]
   El estado (oculto) completo en el tiempo t es $X_t = (X_t^{(1)}, X_t^{(2)})$.

2. **Dinámica de Transición**
   Cada brazo transita **independientemente** del otro. Para el brazo i,
   [
   P_i(x' \mid x) ;=; \Pr\bigl(X_{t+1}^{(i)} = x' ,\big|; X_t^{(i)} = x,; A_t\bigr),
   ]
   donde la acción A_t puede importar si la dinámica del bandido cambia cuando se jala. Estas matrices de transición ${P_i}$ son **conocidas** de antemano.

3. **Observaciones**
   El agente **no** ve X_{t+1}. En cada tiempo t, jala el brazo A_t y **observa la recompensa** R_t. (Podría haber señales adicionales, pero la idea principal es que el estado verdadero está oculto).

4. **Modelo de Recompensa**
   Cuando se jala el brazo A_t, el agente recibe
   [
   R_t ;=; R\bigl(X_t^{(A_t)}\bigr),
   ]

PROF

donde $R(\cdot)$ podría ser Bernoulli u otra función conocida. Como el estado está oculto, el agente solo ve la muestra resultante de la recompensa.

5. **Bono de Exploración**
   En cada paso t, el agente recibe adicionalmente β , B_t. La función B_t podría depender de cuán incierto está el agente sobre los estados futuros o de cuántas veces se ha elegido cada brazo. β es una constante fija.

6. **Objetivo**
   En un horizonte fijo T, se define
   [
   \max_{\text{Pol}} ;\mathbb{E}^{\text{Pol}}!\Bigl[\sum_{t=1}^T \bigl(R_t + \beta,B_t\bigr)\Bigr].
   ]
   Dado que las transiciones son **conocidas** pero el estado está **oculto**, el agente debe mantener una **creencia** sobre $(X_t^{(1)}, X_t^{(2)})$. En principio, se puede establecer un enfoque de programación dinámica parcialmente observable.

Problema Básico 2

Independiente, Transiciones Desconocidas, T Fijo, (Estado Oculto) + Bono

1. **Espacio de Estados**
   Igual que en el Problema Básico 1: $\mathcal{X}^{(i)}$ y $X_t = (X_t^{(1)}, X_t^{(2)})$.

2. **Dinámica de Transición**
   Cada brazo transita de forma independiente, pero ahora las **probabilidades de transición exactas** son desconocidas para el agente. Este comienza con una distribución a priori sobre posibles ${P_i}$ y la actualiza a medida que observa recompensas (y potencialmente cualquier indicio parcial de transiciones, aunque típicamente solo ve recompensa).

3. **Observaciones**
   Nuevamente, el agente **no** observa directamente $(X_{t+1}^{(1)}, X_{t+1}^{(2)})$. Solo observa la recompensa del brazo que eligió. A lo largo del tiempo, usa estas observaciones para inferir tanto los estados ocultos (si es posible) como los parámetros de transición desconocidos.

PROF

4. **Modelo de Recompensa**
   Misma estructura que el Problema Básico 1:
   [
   R_t ;=; R\bigl(X_t^{(A_t)}\bigr).
   ]
   La función de recompensa podría ser conocida o parcialmente conocida. El agente ve únicamente la muestra R_t.

5. **Bono de Exploración**
   β,B_t se añade en cada paso.

6. **Objetivo**
   [

$$\max_{\text{Pol}} \quad \mathbb{E}^{\text{Pol}}!\Bigl[\sum_{t=1}^T \bigl(R_t + \beta,B_t\bigr)\Bigr],$$

donde la esperanza está sobre la incertidumbre del agente en las matrices de transición y los estados ocultos.

Problema Básico 3

Dependiente (Conjunta), Transiciones Conocidas, T Fijo, (Estado Oculto) + Bono

1. **Espacio de Estados**
   $X_t = (X_t^{(1)}, X_t^{(2)})$, con $\lvert \mathcal{X} \rvert = \lvert \mathcal{X}^{(1)} \rvert \times \lvert \mathcal{X}^{(2)} \rvert$.

2. **Dinámica de Transición**
   El estado conjunto evoluciona según
   $$X_{t+1} ;\sim; P\bigl(\cdot \mid X_t,; A_t\bigr),$$
   lo cual **puede** codificar correlaciones entre los dos brazos en los siguientes estados. Esta matriz de transición P es **conocida** desde el inicio.

3. **Observaciones**
   El agente **no** ve X_{t+1}. Solo observa la recompensa del brazo que elija. Por ello, debe mantener una creencia sobre el estado conjunto.

4. **Modelo de Recompensa**
   Misma idea: la recompensa depende del sub-estado del brazo elegido,
   $$R_t ;=; R\bigl(X_t, A_t\bigr).$$

PROF

5. **Bono de Exploración**
   Se añade β,B_t en cada paso.

6. **Objetivo**
   $$\max_{\text{Pol}} \quad \mathbb{E}^{\text{Pol}}!\Bigl[\sum_{t=1}^T \bigl(R_t + \beta,B_t\bigr)\Bigr].$$
   Similar al Problema Básico 1, pero ahora con transiciones **dependientes**. El agente conoce P pero no X_t.

Problema Básico 4

Observabilidad Parcial / Posiblemente Desconocido / Posiblemente T Aleatorio

Este último problema básico puede incorporar **cualquiera** o **todas** las siguientes opciones: las transiciones pueden ser conocidas o desconocidas; el horizonte T puede ser aleatorio o infinito; el estado está oculto (como siempre), y puede haber dinámicas de observación más complejas.

1. **Espacio de Estados**
   $X_t = (X_t^{(1)}, X_t^{(2)})$, oculto para el agente.

2. **Modelo de Observación**
   El agente elige el brazo A_t, luego ve una observación O_t (a menudo solo la recompensa R_t). Las probabilidades de observación pueden ser conocidas o desconocidas. El agente nunca ve directamente el estado.

3. **Dinámica de Transición**

   - Puede ser independiente o conjunta.
   - Puede ser conocida o desconocida.

4. **Estado de Creencia**
   Dado que X_t está oculto, el agente lleva un **estado de creencia** $b_t(x) = \Pr(X_t = x \mid \text{historial hasta }t)$. Esto convierte el problema en un **POMDP**.

5. **Recompensa + Bono**
   [
   R_t ;=; R(X_t, A_t),\quad;;\text{más}\quad \beta,B_t.
   ]

6. **Objetivo**
   Para un **horizonte aleatorio** o **infinito**, se podría tener
   [
   \max_{\text{Pol}} ;\mathbb{E}^{\text{Pol}}!\Bigl[\sum_{t=1}^T \gamma^{t-1}\bigl(R_t + \beta,B_t\bigr)\Bigr],
   ]
   donde $\gamma\le1$ es un factor de descuento, o se mantiene la misma forma finita si T es aleatorio pero conocido en distribución.

PROF

3.3 Conmutación de Conocido vs. Desconocido, Horizontes, etc.

Mostramos ahora cómo **cualquier** combinación de características está cubierta por estas cuatro definiciones básicas:

- **Transiciones Conocidas vs. Desconocidas**

  - Si las transiciones son conocidas, usar Problema Básico 1 o 3.
  - Si las transiciones son desconocidas, usar Problema Básico 2 o una variante de 3 (transiciones conjuntas) con parámetros desconocidos.

- **Independiente vs. Dependiente**

  - **Independiente**: Transiciones factorizadas $P_1\times P_2$.

    - **Dependiente**: Matriz conjunta $P(x_{t+1}\mid x_t, A_t)$.

- **Horizonte Fijo vs. Aleatorio/Infinito**

    - **Fijo**: $\sum_{t=1}^T(R_t + \beta B_t)$.
    - **Aleatorio o Infinito**: $\sum_{t=1}^\infty \gamma^{t-1}(R_t + \beta B_t)$, o un tiempo de parada aleatorio.

- **Observabilidad Parcial**
  En **todos** estos problemas, el agente **no** ve X_t. Debe usar las observaciones (como la recompensa) para mantener una creencia. Surgen más complejidades si el modelo de observación proporciona más o menos información sobre los estados ocultos.

Combinando estos elementos (conocimiento de transiciones, independencia vs. dependencia, tipo de horizonte, etc.), cubrimos todo el espectro de variantes de bandido de Markov de dos brazos—siempre con estados ocultos.

---

4. Guía de Modelado Explícito (Enfoque Heurístico)

Mientras que los **Problemas Básicos** anteriores son plenamente rigurosos, muchos estudiantes se benefician de **heurísticas** o convenciones de nombres simplificados para desarrollar intuición. A continuación, reinterpretamos los mismos problemas de manera más accesible.

4.1 Etiquetado de Estados

En lugar de enumerar estados como ${1,2,3,\ldots}$ o símbolos abstractos, podemos etiquetarlos como:

- Estados **“Bueno”, “Malo”, “Neutral”**, o
- Estados **“Alto”, “Medio”, “Bajo”**.

Aunque el agente **no** ve directamente estos estados, pensarlos así puede ayudar a conceptualizar los procesos de Markov ocultos.

PROF

4.2 Patrones de Transición

- **“Subir”**: El bandido puede pasar de un estado inferior a uno superior (L $\to$ M, M $\to$ H).
- **“Bajar”**: El bandido podría degradarse (H $\to$ M, M $\to$ L).
- **“Quedarse”**: El bandido permanece en el mismo estado.

Con los estados ocultos, el agente solo ve la recompensa, que *podría* correlacionarse con “Bueno” vs. “Malo”.

4.3 Recompensa

Un estado “Bueno” podría proporcionar una recompensa más alta en promedio que un estado “Malo”. Podríamos imaginar un pago Bernoulli con probabilidad de éxito 0.8 para “Bueno” frente a 0.2 para “Malo”, o cualquier otra distribución. El agente infiere estas transiciones y probabilidades de éxito a lo largo del tiempo—especialmente si son inicialmente desconocidas.

4.4 Conexión con los Problemas Básicos

- **Problema Básico 1**: "Conocemos la matriz de transición, pero no el estado real". Es un modelo parcialmente observable con dinámicas de Markov conocidas.
- **Problema Básico 2**: También no vemos el estado, y ahora los parámetros de transición son desconocidos, así que el agente debe aprenderlos.

La **visión heurística** (p. ej., estados "Bueno/Malo") coincide exactamente con la **visión formal** de procesos de Markov con estado oculto (Sección 3), excepto que el conocimiento del agente se limita a observar recompensas.

---

5. Calibración de Parámetros (Especialmente β para Exploración)

Un ingrediente **clave** en bandidos de Markov de dos brazos con bono de exploración es la elección de β. Aquí ofrecemos algunas pautas simples:

1. **Horizonte Corto vs. Largo**
    - **Corto** (T pequeño): El agente tiene menos pasos para beneficiarse de la exploración. Un β *grande* podría ser **contraproducente**, pues se "gasta" en exploración que no se rentabiliza a tiempo.
    - **Largo** (T grande): Un β algo **mayor** puede ayudar a descubrir estados de mayor valor o estimaciones de transición más precisas al principio.
2. **Transiciones Conocidas vs. Desconocidas**
    - Si las transiciones son **plenamente conocidas**, el papel de β podría ser menor dado que el agente no explora para aprender transiciones (aunque puede explorar para "recapturar" o confirmar estados de alto valor).
    - Si las transiciones son **desconocidas**, un β **más grande** ayuda al agente a no caer en la explotación ingenua de solo el brazo aparentemente mejor, fomentando la obtención de información.

PROF

3. **Siempre Estados Ocultos**
    - Como el agente nunca ve el estado real, el **valor de la exploración** suele ser más alto. Un β distinto de cero impulsa al agente a jalar un brazo que podría estar en un mejor estado (oculto) de lo que sugieren las observaciones de recompensa de corto plazo.
4. **Ajuste Empírico**
   A menudo β se ajusta de forma experimental (p. ej., mediante búsqueda en malla/grid search). Los estudiantes pueden realizar simulaciones para ver cómo distintos valores de β afectan el desempeño general.

---

6. Ejercicios

A continuación, se presenta un **nuevo escenario** aplicado a cada uno de los **cuatro problemas básicos** (de las Secciones 3.2–3.3 del texto principal). La **única** modificación es:

> **Al comienzo de cada turno, el agente puede elegir exactamente una máquina y obtener su probabilidad actual de ganar (es decir, la probabilidad de que proporcione un "éxito" o recompensa "1" en ese turno). Tras ver esa probabilidad (solo para una máquina), el agente decide cuál brazo jalar en ese turno.**

Todos los demás aspectos (conocimiento de matrices de transición, estados ocultos u observados, tipo de horizonte, bono de exploración, etc.) siguen como se definieron originalmente en cada problema básico.

Tu **tarea** es **reformular y especificar completamente** cada uno de los cuatro problemas básicos **bajo esta nueva opción de consulta**. Es decir, debes:

1. **Definir el espacio de estados, acciones, transiciones, observaciones y objetivo** exactamente como antes, *pero* con la acción/observación adicional de que en cada turno puedes solicitar la "probabilidad actual de ganar" de una máquina.
2. **Aclarar** cómo encaja esta probabilidad recién revelada en el conocimiento o la actualización de creencias del agente.
3. **Explicar** (de manera breve y rigurosa) cómo el problema sigue siendo el mismo en los demás aspectos (p. ej., qué matrices de transición son conocidas vs. desconocidas, cómo se maneja el horizonte, y cómo se aplica el bono de exploración).

Finalmente, para cada problema, **añade una breve "pregunta de intuición"** que te invite a reflexionar sobre cómo esta pieza extra de información podría influir en la estrategia o el desempeño del agente. **No** se requiere proporcionar ni intentar una solución completa—solo formular el problema revisado y plantear la pregunta de intuición.

Ejercicio 6.1: Problema Básico 1

(Independiente, Transiciones Conocidas, T Fijo)

PROF

- **Puntos Clave Originales**:
    1. El agente conoce las matrices de transición exactas de ambos brazos (transiciones independientes).
    2. El horizonte T es fijo.
    3. Los estados están ocultos, pero el agente conoce cómo evolucionan.
    4. Normalmente, el agente observa solo la recompensa de la máquina elegida en cada turno.
    5. Hay un bono de exploración β,B_t que se añade a la recompensa en cada paso.
- **Nueva Modificación**:
  Al inicio de **cada turno**, el agente puede **elegir exactamente una** de las dos máquinas para conocer su **probabilidad actual de recompensa** (o éxito). Esto ocurre *antes* de que el agente decida cuál brazo jalar ese turno.
- **Reformula el Problema**:

1. **Espacio de Estados / Transiciones**: Igual que antes; el estado de cada brazo evoluciona de manera **conocida** e independiente.
2. **Acciones**: Ahora se dividen en dos fases cada turno:
    - **Fase de Consulta** (opcional pero puede hacerse una vez por turno): El agente especifica el brazo 1 o 2 para "ver" su probabilidad de éxito.
    - **Fase de Elección**: Después, el agente elige qué brazo jalar y recibe la recompensa correspondiente (más el bono de exploración).
3. **Observaciones**:
    - A partir de la **consulta**, el agente ve la probabilidad de éxito del brazo elegido para esa consulta.
    - Al jalar un brazo, el agente observa la recompensa (binaria o de otro tipo).
4. **Objetivo**: Maximizar $\displaystyle \mathbb{E}\bigl[\sum_{t=1}^T (R_t + \beta,B_t)\bigr]$ sobre todas las políticas que pueden usar el resultado de la consulta.

- **Pregunta de Intuición**:
  ¿De qué manera disponer de la probabilidad exacta de éxito de un brazo cada turno influiría en la decisión de qué máquina jalar?

Definición del Problema

Pregunta de Intuición

PROF

Ejercicio 6.2: Problema Básico 2

(Independiente, Transiciones Desconocidas, T Fijo)

- **Puntos Clave Originales**:

    1. Las probabilidades de transición de cada brazo son **desconocidas** al inicio; el agente debe aprenderlas de la experiencia.
    2. El horizonte T es fijo.
    3. Cada brazo evoluciona de manera independiente según una matriz de transición desconocida.
    4. Normalmente, el agente solo ve la recompensa del brazo elegido cada turno.
    5. Se añade β,B_t a la recompensa en cada paso.

- **Nueva Modificación**:
  En cada turno, **antes de jalar** cualquier brazo, el agente puede consultar **una** máquina para saber

su probabilidad actual de éxito. Dicha probabilidad proviene de las transiciones desconocidas, pero se revela en ese instante solo para ese turno.

- **Reformula el Problema**:

    1. **Espacio de Estados / Transiciones**: Misma estructura de estados ocultos, pero las transiciones son desconocidas y deben inferirse.
    2. **Acciones**:
        - **Consultar** el brazo 1 o el brazo 2 para ver su probabilidad de éxito en ese turno.
        - Luego **jalar** un brazo y obtener la recompensa habitual más el bono de exploración.
    3. **Observaciones**:
        - La probabilidad de éxito del brazo consultado.
        - La recompensa del brazo elegido al final del turno.
    4. **Objetivo**: Maximizar $\displaystyle \mathbb{E}\bigl[\sum_{t=1}^T (R_t + \beta,B_t)\bigr]$ mientras se actualizan las creencias sobre las matrices de transición desconocidas, usando potencialmente la consulta para reducir incertidumbre.

- **Pregunta de Intuición**:
  Dado que las transiciones son desconocidas, ¿cómo podría conocer la probabilidad exacta de ganar para un brazo en cada turno acelerar tu estimación de su dinámica de transición—o se usaría principalmente para explotar a corto plazo?

Definición del Problema

Pregunta de Intuición

PROF

Ejercicio 6.3: Problema Básico 3

(Dependiente, Transiciones Conocidas, T Fijo)

- **Puntos Clave Originales**:

    1. El siguiente estado de ambos brazos puede estar **correlacionado**, con una matriz de transición conjunta.
    2. Esta matriz de transición es **conocida**.
    3. El horizonte T es fijo.
    4. El estado oculto $(X_t^{(1)}, X_t^{(2)})$ nunca se observa directamente; el agente solo ve la recompensa del brazo que jala cada turno.
    5. Se añade un bono de exploración β,B_t en cada paso.

- **Nueva Modificación**:
  Al comienzo de cada turno, el agente puede consultar **un** brazo para conocer su probabilidad de éxito actual. Dado que las transiciones están correlacionadas, saber la probabilidad de un brazo podría también afectar la creencia sobre el otro.

- **Reformula el Problema**:

  1. **Espacio de Estados / Transiciones**: La misma cadena de Markov conjunta, **conocida**.
  2. **Acciones**:
     - **Consultar** la probabilidad de éxito de uno de los dos brazos.
     - **Jalar** un brazo y obtener la recompensa más el bono de exploración.
  3. **Observaciones**:
     - La probabilidad de éxito del brazo consultado (lo que podría brindar información parcial sobre el estado conjunto).
     - La recompensa del brazo seleccionado.
  4. **Objetivo**: Maximizar $\displaystyle \sum_{t=1}^T (R_t + \beta,B_t)$ en esperanza, considerando las transiciones conjuntas y la información adicional de la consulta.

- **Pregunta de Intuición**:
  ¿Cómo podría revelar la probabilidad de éxito de un brazo proporcionar información sobre el otro brazo, dada la correlación en las transiciones?

Definición del Problema

Pregunta de Intuición

PROF

Ejercicio 6.4: Problema Básico 4

(Observabilidad Parcial / Posiblemente Desconocido / Posiblemente T Aleatorio)

- **Puntos Clave Originales**:

  1. Es el caso más general: los estados están ocultos, las transiciones pueden ser conocidas o desconocidas, y el horizonte puede ser aleatorio o infinito.
  2. Típicamente, el agente solo observa la recompensa del brazo elegido (u otra observación parcial).
  3. Se sigue añadiendo β,B_t en cada paso, y las creencias deben actualizarse al estilo **POMDP**.

- **Nueva Modificación**:
  En cada turno, el agente puede consultar **un** brazo para saber su probabilidad de éxito actual. Esta observación adicional podría reducir la observabilidad parcial en cada turno, pero solo para el brazo consultado.

- **Reformula el Problema**:

  1. **Espacio de Estados / Transiciones**: Los estados ocultos y transiciones siguen como en el Problema 4 original. Pueden ser conocidos o desconocidos, posiblemente correlacionados, con horizonte aleatorio o infinito.
  2. **Acciones**:
     - **Consultar** la probabilidad de éxito de un brazo.
     - **Jalar** el brazo 1 o 2, observando la recompensa más el bono de exploración.
     - Posiblemente continuar hasta un tiempo de parada aleatorio o indefinidamente (si el horizonte es infinito).
  3. **Observaciones**:
     - Si las transiciones son conocidas, la consulta ayuda a localizar el estado oculto.
     - Si son desconocidas, ayuda a refinar los parámetros o estados de creencia.
  4. **Objetivo**: **Maximizar** la recompensa acumulada esperada (más los bonos) en un horizonte aleatorio o infinito, considerando la observabilidad parcial y la nueva acción de consulta.

- **Pregunta de Intuición**:
  ¿Cómo prevés usar la consulta de probabilidad de éxito de un brazo en un entorno POMDP o con horizonte aleatorio? ¿Te permitiría explotar con más confianza un brazo prometedor o sería principalmente valiosa para refinar tu creencia a lo largo del tiempo?

Definición del Problema

PROF **Pregunta de Intuición**

---

7. Algoritmos de Inferencia de Estados y Decisión

*(Continuación del Documento Original, con las secciones previas intactas; **únicamente** se reubican las definiciones de símbolos en las secciones donde se emplean.)*

En esta sección presentamos los **algoritmos principales** para:

1. **Inferir** (estimar) los estados subyacentes —y, en algunos casos, las **transiciones** y/o parámetros — de procesos de Markov **ocultos** (Hidden Markov Chains) asociados a los bandidos de dos brazos.
2. **Decidir** qué acción tomar para **maximizar** el rendimiento en escenarios de **exploración vs. explotación**, incluyendo variantes como UCB (Upper Confidence Bound), Thompson Sampling, políticas ε-Greedy y bonos de exploración.

Estos métodos están directamente relacionados con los **Problemas de Decisión Básicos** (Sección 3 del documento original), donde el agente:

- No observa directamente el estado X_t.
- Recibe recompensas ${R_t}$ (y, opcionalmente, otra señal) dependiendo de la acción elegida.
- Debe decidir cómo jalar cada brazo para optimizar la **ganancia total** a lo largo de un horizonte (fijo, aleatorio o infinito).

La presentación mantiene el **estilo formal** del documento, complementándola con la descripción de los símbolos en los lugares pertinentes para **no** duplicar la información. Al final (Sección **8.6**) se describe cómo se **integran** los algoritmos de inferencia y los de decisión en el ciclo de un bandido de Markov.

7.1. Visión General de la Inferencia y Decisión

Recordemos que en los problemas de bandidos de dos brazos **con estado oculto**:

- El estado en el tiempo t se denota $X_t = (X_t^{(1)}, X_t^{(2)})$, o únicamente $X_t^{(i)}$ por brazo cuando son independientes.
- Las **transiciones** están dadas por una matriz (o matrices)
  $
  P(x' \mid x, A) = \Pr(X_{t+1} = x' \mid X_t = x, A_t = A),
  $
  donde $A\in{1,2}$ indica qué brazo se jala.
- El agente **observa** recompensas R_t (y quizá alguna otra señal, como la "probabilidad de éxito" si la consulta está permitida).
- El agente mantiene o actualiza una **creencia** sobre X_t, y puede además actualizar su conocimiento de los parámetros de transición si estos son desconocidos.
- Con dicha creencia, selecciona una **acción** $A_t \in {1,2}$ para maximizar la utilidad (recompensa y/o bono de exploración).

PROF

Notación General (usada en toda la Sección 8)

1. **X_t**: Estado (oculto) del sistema en tiempo t.
2. **A_t**: Acción en tiempo t ($\in {1,2}$).
3. **R_t**: Recompensa observada al jalar el brazo A_t. Con frecuencia es Bernoulli, pero puede ser real.
4. **O_t**: Observación en tiempo t (en la forma más simple, coincide con R_t).
5. **$\pi_0(x)$**: Distribución inicial sobre estados, $\Pr(X_1 = x)$. Es un **vector** en $\mathbb{R}^{|\mathcal{X}|}$ con $\sum_x \pi_0(x) = 1$.

6. **$b_t(x)$**: Creencia (distribución posterior) sobre X_t tras ver $O_{1:t}$ y $A_{1:t-1}$. Otro **vector** de dimensión $\lvert \mathcal{X}\rvert$.
7. **$P(x' \mid x, A)$**: Matriz (o familia de matrices) de transición. Cada matriz es $\lvert \mathcal{X}\rvert \times \lvert \mathcal{X}\rvert$.
8. **π** (distribución estacionaria): Satisface $\pi = \pi P$, de dimensión $\lvert\mathcal{X}\rvert$.
9. **$\hat{P}$**, **$\hat{\pi}_0$**: Estimaciones (gorrito) de matrices de transición o distribución inicial, respectivamente, cuando son **desconocidas** y aprendidas.
10. **θ**: Parámetro(s) en un enfoque bayesiano (p.ej. la dinámica).
11. **β**, **ε**: Parámetros que controlan la exploración (bono de exploración y ε-Greedy).
12. **$B_t(i)$**: Bono de exploración para la acción i.
13. **$\overline{r}_i$, $n_i(t)$**: En UCB clásico (i.i.d.), $\overline{r}_i$ es la media observada del brazo i, y $n_i(t)$ es su conteo de tirones hasta tiempo t. Para Markov oculto puede adaptarse con estimadores que consideren el estado.

En lo que sigue, cada algoritmo puede introducir *variables adicionales* (por ejemplo, $\delta_t(x)$, $\psi_t(x)$ en Viterbi).

7.2. Algoritmos de Inferencia de Estados

En los bandidos de Markov **ocultos**, el agente no observa X_t directamente. Para estimar sus valores (o, más estrictamente, su **distribución**), se usan métodos de **Hidden Markov Models (HMM)**.

7.2.1. Algoritmo Forward

(Filtrado de Creencias en HMM)

1. **Objetivo**
   Calcular la **probabilidad posterior** (o creencia) sobre el estado en el tiempo t:
   $$
   b_t(x) = \Pr(X_t = x \mid O_{1:t}, A_{1:t-1}).
   $$
   Esto se llama "filtrado" en HMMs.

PROF

2. **Inputs**
   - Distribución inicial del estado $b_1(x) = \pi_0(x)$.
   - Matriz de transición **conocida** $P(x'\mid x,A)$.
   - Modelo de observación $\Pr(O_t \mid X_t, A_t)$.
   - Historial ${A_1,\dots,A_{t-1}}$, ${O_1,\dots,O_t}$.

3. **Outputs**
   - Un **vector** b_t donde cada componente $b_t(x)$ da la probabilidad actual de que $X_t=x$.

4. **Fórmula General**

- **Predicción**:
  $
  \tilde{b}*t(x') = \sum*{x} b_{t-1}(x) P(x' \mid x, A_{t-1}).
  $
- **Actualización**:
  $
  b_t(x') = \frac{\Pr(O_t \mid x', A_t)\tilde{b}*t(x')}{\sum*{x''} \Pr(O_t \mid x'',A_t)\tilde{b}_t(x'')}.
  $

5. **Intuición**

   - Primero, el **prediction step** propaga la creencia mediante P.
   - Luego, el **update step** incorpora la observación O_t.

6. **Cuándo se Usa**

   - Cada turno, si el agente quiere una **estimación online** de X_t.
   - Muy útil en **Problema Básico 1**, **3** y en la parte de **4** (cuando las transiciones se conocen o se han estimado).

7. **Relación con Decisión**

   - Se usa $b_t(x)$ para calcular $\mathbb{E}[R_t\mid A]$, integrando sobre la creencia.
   - Aporta la "probabilidad de que cada estado ocurra" al momento de decidir.

7.2.2. Algoritmo Viterbi

(Secuencia de Estados Más Probable)

1. **Objetivo**
   Hallar la **trayectoria** $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T$ que maximiza $\Pr(X_1=\hat{x}*1,\dots, X_T=\hat{x}T \mid O{1:T},A{1:T})$.

PROF

2. **Inputs**

   - Matriz de transición $P(x' \mid x,A)$.
   - Modelo de observación $\Pr(O_t \mid X_t, A_t)$.
   - Secuencia completa ${A_1,\dots,A_T}$ y ${O_1,\dots,O_T}$.
   - Distribución inicial $\pi_0(x)$.

3. **Outputs**

   - Una **secuencia** $\hat{x}_1,\hat{x}_2,\dots,\hat{x}_T$ de estados más probable.

4. **Variables Auxiliares**

   - $\delta_t(x)$: probabilidad (o puntuación) **máxima** de cualquier trayectoria que termine en x al tiempo t.
   - $\psi_t(x)$: estado predecesor que optimiza esa trayectoria.

5. **Fórmula General**

    - Recursión:
      $
      \delta_t(x') = \Bigl[\max_{x} \delta_{t-1}(x) P(x'\mid x,A_{t-1})\Bigr]\times\Pr(O_t \mid x', A_t).
      $
    - $\psi_t(x')$ se actualiza con el "$\arg\max$".

6. **Intuición**

    - Útil para análisis **offline** o diagnóstico de la **trayectoria** más probable.
    - No mantiene la distribución de X_t, sino una sola ruta.

7. **Cuándo se Usa**

    - Para entender ex post "qué ocurrió con el bandido".
    - Menos frecuente en decisión online.

7.2.3. Algoritmo Baum-Welch

(Aprendizaje de Parámetros Desconocidos via EM)

1. **Objetivo**
   Estimar las **matrices de transición** (y el modelo de observación) cuando son **desconocidas**, es decir, obtener $\hat{P}$, $\hat{O}$, etc.

2. **Inputs**

    - Secuencia de acciones ${A_1,\dots,A_T}$.
    - Secuencia de observaciones ${O_1,\dots,O_T}$.
    - Número de estados $\lvert \mathcal{X}\rvert$.
    - Distribución inicial $\pi_0(x)$ (o se estima también).

PROF

3. **Outputs**

    - $\hat{P}(x'\mid x,A)$, $\hat{O}(O_t\mid X_t,A_t)$ que maximizan la verosimilitud de los datos.
    - A veces $\hat{\pi}_0$.

4. **Esquema EM**

    - **E-step**: se usa un forward-backward modificado para calcular las probabilidades esperadas de cada transición/estado.
    - **M-step**: se actualizan $\hat{P}$ y $\hat{O}$ para maximizar la verosimilitud esperada.

5. **Intuición**

    - Itera entre "inferir la probabilidad de cada estado con la conjetura actual" y "actualizar la conjetura de $\hat{P}$".

6. **Cuándo se Usa**
    - En **Problema Básico 2** o **4**, donde no se conocen a priori las transiciones.
    - Puede usarse offline o en forma incremental "online EM".
7. **Relación con Decisión**
    - Con $\hat{P}$ y $\hat{\pi}_0$ aprendidas, se hace luego un filtrado (Alg. Forward) para la acción.
    - O integrarlo dentro de un esquema de decisión (p.ej. actualizando $\hat{P}$ tras cada episodio).

7.2.4. Estimación de Estado Inicial

1. **Objetivo**
    - Determinar o aproximar $\pi_0(x) = \Pr(X_1 = x)$.
2. **Inputs**
    - Datos iniciales (observaciones tempranas, episodios anteriores).
    - A veces parte del Baum-Welch si también se desconoce π_0.
3. **Outputs**
    - Una **distribución** $\hat{\pi}_0(x)$.
4. **Intuición**
    - Importante en horizontes cortos.
    - En horizontes largos, su impacto se reduce.
5. **Relación con Decisión**
    - Inicializa el **Algoritmo Forward** o Viterbi en el primer paso.
    - Puede cambiar acciones tempranas.

PROF

7.2.5. Cálculo de Estado Estacionario (Steady State)

1. **Objetivo**
   Encontrar π tal que $\pi = \pi P$. Es el **autovector** izquierdo asociado al autovalor 1 de P.
2. **Inputs**
    - Matriz de transición P, irreducible y aperiódica.
3. **Outputs**
    - $\pi\in\mathbb{R}^{|\mathcal{X}|}$, con $\sum_x \pi(x)=1$.
4. **Intuición**

- Describe la **frecuencia a largo plazo** de estados si la política es estacionaria y la cadena converge.

5. **Relación con Decisión**
   - Útil en horizonte infinito o para heurísticas de "valor promedio" de cada estado.

7.3. Algoritmos de Decisión (Exploración-Explotación)

Una vez que se dispone (o se actualiza) la información sobre el proceso de Markov (creencia b_t, matrices estimadas $\hat{P}$, etc.), hay que **decidir** cuál brazo jalar:

$$
A_t \in {1,2}.
$$

El objetivo (Sección 3) es maximizar la **recompensa total** (posiblemente con bono de exploración) en un cierto horizonte.

7.3.1. UCB (Upper Confidence Bound)

1. **Objetivo**
   - Mantener, para cada brazo i, un **intervalo de confianza** sobre su valor esperado.
   - Elegir el brazo con el **límite superior** más alto en cada turno.

2. **Inputs**
   - Historial de recompensas o estimaciones de recompensas (en Markov oculto, usando b_t o $\hat{P}$).
   - Alguna forma de computar la **cota superior**.

3. **Outputs**
   - Acción $A_t = \arg\max_{i\in{1,2}} [\text{UCB}_t(i)]$.

PROF

4. **Fórmula Clásica (i.i.d.)**
   $$
   \text{UCB}_t(i) = \overline{r}_i + \sqrt{\frac{2\ln t}{n_i(t)}}.
   $$
   Donde $\overline{r}_i$ es la media observada del brazo i y $n_i(t)$ su conteo.
   En un escenario Markov se reemplazan por estimadores que consideren las probabilidades de estados buenos/malos.

5. **Intuición**
   - Método "optimista": se prueban brazos con alta **cota superior**.
   - Evita descartar brazos poco explorados.

6. **Relación con Inferencia**

- Se pueden combinar con Alg. Forward (para $\mathbb{E}[R_t\mid i]$) y luego añadir un "extra" por la incertidumbre.

7.3.2. Políticas con Bono de Exploración βB_t

1. **Objetivo**
   - **Modificar** la recompensa inmediata añadiendo βB_t, donde B_t mide la "utilidad de información" o la "incertidumbre" de jalar un cierto brazo.
2. **Inputs**
   - Una **función** $B_t(i)$.
   - $\beta\in \mathbb{R}_{+}$ que controla la importancia de la exploración.
3. **Outputs**
   $
   A_t = \arg\max_{i \in {1,2}} \Bigl[\mathbb{E}[R_t \mid i] + \beta B_t(i)\Bigr].
   $
4. **Intuición**
   - Similar a UCB, pero en lugar de cota superior, se agrega un bono explícito.
   - β ajusta "cuánto" se incentiva la exploración.
5. **Relación con Inferencia**
   - $\mathbb{E}[R_t \mid i]$ puede venir de b_t.
   - $B_t(i)$ depende de cuán incierta sea la creencia respecto al estado o parámetros del brazo i.

7.3.3. ε-Greedy

PROF

1. **Objetivo**
   - Con prob. $1-\varepsilon$, actuar **greedy**; con prob. ε, **explorar** (elegir una acción al azar).
2. **Inputs**
   - $\varepsilon\in(0,1)$.
   - $\mathbb{E}[R_t\mid i]$ (vía inferencia) o alguna estimación.
3. **Outputs**
   $
   A_t =
   \begin{cases}
   \arg\max_{i} \mathbb{E}[R_t \mid i], & \text{con prob }1-\varepsilon,\
   \text{(un brazo al azar)}, & \text{con prob }\varepsilon.

\end{cases}
$

4. **Intuición**

    - Sencillo de implementar. Garantiza explorar algo.
    - Exploración no "dirigida".

5. **Relación con Inferencia**

    - Para la parte greedy, $\mathbb{E}[R_t \mid i]$ se obtiene del filtrado (Forward) o de $\hat{P}$.

7.3.4. Thompson Sampling

1. **Objetivo**

    - Enfoque **bayesiano**: se mantiene una posterior $\Pr(\theta \mid \text{historial})$ sobre parámetros (probabilidades de transición, etc.).
    - En cada paso, se **samplea** $\theta^$ *y se elige la acción que optimiza la recompensa esperada dada $\theta^$*.

2. **Inputs**

    - θ: descripción paramétrica de la dinámica del bandido.
    - Observaciones pasadas para mantener la posterior.

3. **Outputs**

    - Acción A_t.
    - Implícitamente, una muestra θ^* de la posterior.

4. **Procedimiento**
$$
\theta^* \sim \Pr(\theta\mid \text{historial}), \quad
A_t = \arg\max_{i} \bigl[\mathbb{E}[R_t \mid \theta^*, i]\bigr].
$$

5. **Intuición**

    - Explora "automáticamente": si un brazo es poco explorado, la varianza alta en θ a veces lo hace parecer muy bueno.

6. **Relación con Inferencia**

    - Para actualizar $\Pr(\theta\mid \text{historial})$, puede usarse un filtro de partículas HMM o un EM bayesiano.
    - Muy usado en **Problema Básico 2** y 4.

7.3.5. Políticas Basadas en Índices (p. ej., Gittins)

1. **Objetivo**
    - Asignar a cada brazo un **índice** (valor futuro + valor presente) y elegir el de índice mayor.
2. **Inputs**
    - Modelo para calcular $\text{Índice}_t(i)$.
3. **Outputs**
   $
   A_t = \arg\max_{i\in{1,2}} \text{Índice}_t(i).
   $
4. **Intuición**
    - En bandidos i.i.d. (no Markov) el **Índice de Gittins** es una solución elegante y óptima (horizonte infinito).
    - En Markov oculto, la extensión es compleja.
5. **Relación con Inferencia**
    - Requiere un modelo para "valorizar" la información.
    - Poco común en la práctica de grandes HMMs por complejidad.

7.4. Relación de Estos Algoritmos con los Problemas de Decisión Básicos

Recordemos la clasificación de la Sección 3:

1. **Problema Básico 1**: Independiente, Transiciones **Conocidas**, Estado Oculto, Horizonte Fijo.
    - **Inferencia**: Algoritmo Forward para filtrar (también se podría Viterbi a posteriori).
    - **Decisión**: Cualquier política que use $\mathbb{E}[R_t \mid i]$ derivada de la creencia. Ej.: UCB (adaptado), ε-Greedy, Políticas con Bono βB_t.

PROF

2. **Problema Básico 2**: Independiente, Transiciones **Desconocidas**, Estado Oculto, Horizonte Fijo.
    - **Inferencia**: Hay que aprender $\hat{P}$ (Baum-Welch o bayesiano) y filtrar estado si hace falta.
    - **Decisión**: Thompson Sampling es natural, o UCB con estimaciones $\hat{P}$.
3. **Problema Básico 3**: **Dependiente** (Conjunta), Transiciones **Conocidas**, Horizonte Fijo, Estado Oculto.
    - **Inferencia**: Forward/Viterbi en el **estado conjunto**.
    - **Decisión**: Igual que (1), pero con mayor complejidad de estado y posible correlación.
4. **Problema Básico 4**: **Observabilidad Parcial / Desconocido / Posiblemente Horizonte Aleatorio**.
    - El más **general**: Baum-Welch + Forward, o métodos bayesianos.

- **Decisión**: Thompson, UCB, etc., adaptados a horizonte infinito o aleatorio.

7.5. Conclusión y Comentarios Finales

Los problemas de **Bandidos de Markov de Dos Brazos con Estado Oculto** demandan la combinación de:

1. **Algoritmos de Inferencia** (p.ej., Forward, Baum-Welch) para manejar la incertidumbre sobre estados o parámetros de transición.
2. **Algoritmos de Decisión** (UCB, Thompson, ε-Greedy, etc.) que balanceen la exploración vs. explotación.

En la práctica, la **eficacia** de cada esquema depende de:

- El **horizonte** (corto vs. largo).
- El **tamaño y complejidad** del espacio de estados.
- Si las **transiciones** son conocidas o no.
- Si la **recompensa** es Bernoulli o más compleja.
- Posibles **consultas** adicionales (ver ejercicios).

No existe "un solo algoritmo" óptimo en todos los casos. A menudo se aplican **heurísticas** que combinan un filtrado (Forward) con un método de decisión como Thompson o UCB, o se practica un **EM online** para ajustar $\hat{P}$ y luego se usa la política con bono de exploración.

7.6. Integración de la Inferencia y la Decisión

Para finalizar, describimos de forma **explícita** cómo se **conectan** los algoritmos de inferencia y los de decisión en la rutina de un **bandido de Markov**:

1. **Inicialización**:
    - Definir π_0.
    - Si las transiciones son desconocidas, inicializar $\hat{P}$ (p.ej. con conjetura uniforme) o la posterior $\Pr(\theta)$.

PROF

2. **Para cada turno** $t = 1, 2, \dots, T$:
    1. **Inferencia (Estado / Parámetros)**
        - Si $\hat{P}$ es **fija** y conocida, se aplica **Forward** con la observación O_{t-1} para obtener b_t.
        - Si $\hat{P}$ es **desconocida**, se hace un update **bayesiano** o un paso de **EM online** para refinar $\hat{P}$.
        - Calcular $\mathbb{E}[R_t \mid A=i]$ con $b_t(x)$.
        - (Opcional) Calcular "incertidumbre" para UCB o B_t.
    2. **Decisión (Exploración-Explotación)**
        - Escoger A_t (UCB, Thompson, ε-greedy, bono, etc.).

- Ejecutar A_t.

3. **Observación y Recompensa**
    - Observar O_t (p. ej., R_t).
    - Incorporar O_t en el paso de inferencia del siguiente turno.

3. **Repetir** hasta T (o indefinidamente, si infinito).

Este **bucle** deja claro que la **Inferencia** (cálculo de $\hat{P}$, b_t, etc.) provee la "materia prima" para la **Decisión**, y la acción produce nuevas **observaciones** para refinar la inferencia. Así se cumple el ciclo de **Aprendizaje por Refuerzo** en un **entorno de Markov Oculto**.

---

8. Resolución del Bandido Markoviano como Problema de Decisión

A la hora de **resolver** un **bandido markoviano** como **problema de decisión**, el **razonamiento algorítmico** se compone de varias **capas** que van desde la **representación formal** de la dinámica y las recompensas, hasta la **estrategia** para balancear la exploración y la explotación. A continuación, se describe *cómo* estructurar y **pensar** este proceso de manera sistemática, tomando como base el planteamiento de un **problema de decisión**.

8.1. Representar Formalmente el Problema de Decisión

Lo primero es traducir el escenario de bandido (dos brazos) al lenguaje de **Procesos de Decisión** (sea MDP o POMDP). Esto implica:

1. **Definir el conjunto de estados** $\mathcal{X}$.
    - Cuando hay **estado oculto**, $\mathcal{X}$ contiene los valores posibles del sistema subyacente.
    - Si cada brazo puede estar en "Bueno/Malo", el estado global puede ser ${(\text{B}, \text{B}), (\text{B}, \text{M}), \dots}$.

PROF

2. **Definir el conjunto de acciones** $\mathcal{A}$.
    - En un **bandido de dos brazos**, $\mathcal{A} = {1, 2}$: elegir el brazo 1 o el brazo 2.

3. **Definir la función de transición** $P(x_{t+1} \mid x_t, a_t)$.
    - Esta describe cómo evoluciona el estado al tomar la acción a_t.
    - Puede haber independencia (cada brazo evoluciona por su cuenta) o dependencia conjunta.

4. **Definir la función de observación** $\Pr(o_t \mid x_t,a_t)$.
    - Si el estado es oculto, solo vemos las **recompensas** (y/o señales) que dependen de x_t y de la acción.

5. **Definir la función de recompensa** $R(x_t, a_t)$.

- En un bandido Bernoulli, $R(x,a)$ es la probabilidad de "éxito" al elegir el brazo a en el estado x.
- Si el problema incluye un **bono de exploración**, puede sumarse al reward: $R'(x,a) = R(x,a) + \beta ,B(\text{incertidumbre})$.

6. **Establecer el horizonte** (fijo T o infinito) y la **función objetivo**.
   - Por ejemplo, maximizar $\mathbb{E}\bigl[\sum_{t=1}^T R(X_t, A_t)\bigr]$.

En problemas con **estado oculto**, la representación final es un **POMDP**: $(\mathcal{X}, \mathcal{A}, \mathcal{O}, P, \Omega, R)$, donde Ω es el modelo de observación y $\mathcal{O}$ el espacio de observaciones.

8.2. Identificar las Variables de Decisión y el Flujo de Información

Una vez tenemos la representación:

1. **Información del agente**:
   - Si es oculto, el agente mantiene una **creencia** b_t.
   - En cada paso, ve una **observación** (o_t o r_t) y **actualiza** su creencia mediante filtrado.
2. **Acción**
   - La **decisión** consiste en escoger $A_t \in {1,2}$, basándose en la creencia b_t o en los parámetros estimados.
3. **Resultado**
   - Se obtiene una recompensa (y un nuevo **estado**, aunque oculto), que genera la **nueva observación**.

En el **pensamiento algorítmico** del **problema de decisión**, uno se enfoca en:

PROF

- **Qué** información tienes al inicio de cada turno (creencia, estimaciones).
- **Qué** acción tomar (el *control*).
- **Cómo** evoluciona el sistema y qué observación llega.

8.3. Construir un Mecanismo de Inferencia (si hay Estado Oculto)

Dado que es un bandido **markoviano oculto**, necesitamos un paso de **inferencia**:

1. **Creencia** b_t en el tiempo t.
2. **Algoritmo** (Forward) para pasar de b_{t-1} a b_t:
   $$
   b_t(x) = \frac{\Pr(o_t \mid x,,a_{t-1}); \sum_{x'} b_{t-1}(x'),P(x\mid x',a_{t-1})}{\dots}.
   $$

Si las **transiciones** son desconocidas, se combina con un **aprendizaje de parámetros** (Baum-Welch, un posterior bayesiano, etc.). Este **bloque de inferencia** provee la "materia prima" de la **decisión**: la

probabilidad (o distribución) de estar en cada estado.

8.4. Plantear la Búsqueda de la Política Óptima

La parte central del **problema de decisión** es encontrar la **política** que maximice la suma de recompensas esperadas (más, opcionalmente, un bono de exploración). Los caminos principales para "pensar" el algoritmo de decisión son:

8.4.1. Resolver la Programación Dinámica (PD) o POMDP (Value Iteration)

- **Value Iteration** en el **espacio de creencias** (POMDP):
  $$
  V_t(b) = \max_{a}\Bigl[,R(b,a);+;\sum_{o} \Pr(o\mid b,a),V_{t-1}(\tau(b,a,o))\Bigr].
  $$
  - Requiere discretizar o aproximar el espacio $\mathcal{B}$.
  - Óptimo, pero costoso.
- **Policy Iteration** (también en $\mathcal{B}$), usando α-vectores, PBVI, etc.

En un bandido de dos brazos con pocos estados, esto es *factible*, pero puede ser arduo si $\mathcal{X}$ crece.

8.4.2. Emplear Heurísticas de Exploración-Explotación

- **Greedy + Bono**:
  - Se computa $\mathbb{E}[R_t \mid a]$ y se suma un término $\beta ,B_t(a)$.
  - Esto no garantiza la optimalidad global, pero sí un enfoque computacionalmente más simple.
- **Thompson Sampling**:
  - Mantener posterior sobre la dinámica (o sobre la recompensa por estado).
  - Samplear $\theta^$ *y actuar de forma optimista si la muestra θ^** sugiere un brazo con mayor probabilidad de premio.
- **UCB** adaptado:
  - Confiar en una cota superior $\text{UCB}_t(a)$ que englobe la incertidumbre; elegir aquel con mayor UCB.

Estas heurísticas *indirectamente* incorporan la valoración futura de la información, pero sin resolver la PD exacta.

8.5. Insertar la Estructura del Ciclo de Decisión

La **arquitectura** de un algoritmo final (que orquesta todo) queda así:

1. **Inicialización**:
    - Establecer creencia inicial $b_1(x)$ (o π_0).
    - Si no se conocen transiciones, inicializar su estimación (ej. $\hat{P}$ con algo uniforme).
2. **Para** $t=1$ hasta T (o de forma indefinida):
    1. **Inferencia**:
        - Dada la creencia b_{t} (o si es el paso 1, usar b_1), y la recompensa/observación previa, se hace el **update** al ver o_t.
        - Si las transiciones son desconocidas, se ajustan mediante recuentos, Baum-Welch, o posterior bayesiano.
    2. **Decisión**:
        - Con la creencia (o la muestra θ^*), se elige A_t según:
          $$
          A_t = \arg\max_a \Bigl(\mathbb{E}[R_t \mid a] + \text{exploración}\Bigr).
          $$
        - Ejemplos: UCB, Thompson, ε-Greedy, Value Iteration POMDP, etc.
    3. **Ejecutar Acción y Observar**:
        - Se tira el brazo A_t, se obtiene R_t (y cualquier señal extra).
        - Se usará en el siguiente paso de **inferencia**.
3. **Salida**:
    - En horizonte fijo T, al terminar se contabiliza la **suma** de recompensas; en infinito, se mide la ganancia promedio/descontada.

8.6. Refinamientos del Pensamiento Algorítmico

1. **Analizar la complejidad**:
    - ¿Cuántos estados hay?
    - ¿Es factible la resolución exacta con POMDP?
    - ¿Necesito heurísticas?
2. **Decidir el grado de "optimalidad" necesario**:
    - ¿Busco garantía teórica? (POMDP ex. Value Iteration)
    - ¿Busco escalabilidad? (Heurísticas)
3. **Incorporar aspectos prácticos**:
    - Ajuste de β en el bono, o ε en ε-Greedy.
    - Estimaciones y contadores para la dinámica, si es desconocida.
    - Mecanismos de early-stopping en EM o regularización.

8.7. Resumen de la Lógica de "Problema de Decisión"

PROF

- Se concibe todo como una **dinámica** que evoluciona en el tiempo, con un **agente** que elige acciones.
- Cada elección de acción afecta la **evolución** del estado y la **observación** de la recompensa.
- Se define un **criterio** de optimalidad (suma de recompensas esperadas, a menudo con exploración) y se **busca** la política que lo maximiza.
- El **Pensamiento Algorítmico** así integra:
    1. **Modelado** de transiciones/observaciones,
    2. **Inferencia** (en caso de ocultamiento del estado),
    3. **Planificación o Decisión** (valor futuro vs. recompensa inmediata),
    4. **Actualización** (la acción genera nueva info que refina la creencia o los parámetros),
    5. **Iteración** hasta agotar el horizonte.

En un **bandido** en particular, gran parte de la **complejidad** recae en:

- Manejar la **incertidumbre** sobre estados o transiciones (sobre todo si son desconocidas).
- Equilibrar "explotar lo que parece mejor" vs. "explorar brazos que podrían ser mejor de lo que aparentan".

Ésa es la clave del razonamiento algorítmico en un problema de decisión con bandidos de Markov:

- **Formulación** clara (estados, acciones, observaciones, recompensas),
- **Inferencia** del estado (o parámetros) si es oculto,
- **Decisión** (POMDP exacto o heurístico),
- **Ciclo** de realimentación (observación actualiza creencias y da paso a la siguiente acción).

9. Guía de Implementación y Uso del Framework de Simulación

Esta sección proporciona una descripción detallada del framework de simulación para los problemas de bandidos markovianos presentados en las secciones anteriores. El objetivo es orientar a los estudiantes en la implementación de agentes que resuelvan los cuatro problemas básicos definidos en la Sección 3.

PROF

9.1. Estructura General del Framework

El framework de simulación está diseñado siguiendo una arquitectura modular que separa:

1. **Entornos** (`MarkovBanditEnvironment`): Implementan las dinámicas de los cuatro problemas básicos.
2. **Agentes** (funciones como `problem1_agent`, `problem2_agent`, etc.): Toman decisiones basadas en la información visible.
3. **Funciones de experimentación** (`run_experiment`, `run_standard_experiment`): Coordinan la interacción entre agentes y entornos.
4. **Funciones de visualización** (`visualize_simplified_results`, etc.): Presentan resultados y métricas.
5. **Funciones de análisis** para estudiar el efecto del bono de exploración.

El flujo de trabajo general consiste en:

- Definir un agente que implemente una estrategia
- Ejecutar experimentos con dicho agente en uno o varios entornos
- Analizar las métricas de rendimiento resultantes

9.2. Clases de Entorno

Cada uno de los problemas básicos está implementado como una clase que hereda de `MarkovBanditEnvironment`:

9.2.1. Problem1Environment

Problema 1: Independiente, Transiciones Conocidas, T Fijo, Estado Oculto + Bono de Exploración

- El estado de cada brazo evoluciona de manera independiente según matrices de transición conocidas.
- Los estados están ocultos para el agente, pero las matrices de transición se proporcionan.
- El horizonte T es fijo y conocido.

9.2.2. Problem2Environment

Problema 2: Independiente, Transiciones Desconocidas, T Fijo, Estado Oculto + Bono de Exploración

- Similar al Problema 1, pero las matrices de transición no se proporcionan al agente.
- El agente debe inferir la dinámica a partir de las recompensas observadas.

9.2.3. Problem3Environment

Problema 3: Dependiente (Conjunta), Transiciones Conocidas, T Fijo, Estado Oculto + Bono de Exploración

- Los estados de ambos brazos evolucionan de manera conjunta (correlacionada).
- Se proporciona la matriz de transición conjunta.
- El agente debe entender cómo la acción en un brazo puede revelar información sobre el otro.

PROF

9.2.4. Problem4Environment

Problema 4: Observabilidad Parcial / Posiblemente Desconocido / Posiblemente T Aleatorio

- La versión más general, que puede combinar cualquiera de las características anteriores.
- Las transiciones pueden ser conocidas o desconocidas.
- El horizonte puede ser fijo o aleatorio.
- Las dinámicas pueden ser independientes o conjuntas.

9.3. Funciones de Agente a Implementar

Los estudiantes deben implementar las siguientes funciones de agente:

```
def problem1_agent(env_info: Dict) -> int:
    """
```

```
    Agente para el Problema 1: Independiente, Transiciones Conocidas, T
Fijo, Estado Oculto + Bono de Exploración

    Este agente debe implementar una actualización de creencia sobre el
estado utilizando
    las matrices de transición conocidas y combinar explotación con
exploración.

    Args:
        env_info (Dict): Diccionario con:
            - current_turn (int): Turno actual (comienza en 0)
            - total_turns (int): Número total de turnos en el juego (T
fijo)
            - transition_matrices (List[np.ndarray]): Matrices de
transición conocidas para ambos brazos
            - state_rewards (List[float]): Recompensas asociadas a cada
estado
            - history (Dict): Historial con:
                - 'actions' (List[int]): Acciones anteriores (0 para
brazo 1, 1 para brazo 2)
                - 'rewards' (List[float]): Recompensas recibidas
                - 'exploration_bonus' (List[float]): Bonos de
exploración recibidos

    Returns:
        int: La acción a tomar (0 para brazo 1, 1 para brazo 2)
    """
    # Implementar agente

    # Ejemplo: elección aleatoria (subóptima)
    return np.random.randint(0, 2)
```

Las otras funciones (`problem2_agent`, `problem3_agent`, `problem4_agent`) siguen un patrón similar, con diferencias en la información disponible en `env_info` según el problema.

PROF

9.4. Métricas de Evaluación

El framework calcula varias métricas para evaluar el rendimiento de un agente:

1. **Recompensa Media**: Promedio de recompensa por turno.
2. **Porcentaje de Acciones Óptimas**: Porcentaje de veces que el agente eligió la acción que maximiza la recompensa esperada.
3. **Arrepentimiento (Regret)**: Diferencia entre la recompensa óptima posible y la recompensa obtenida.
4. **Bono de Exploración**: Suma total de bonos de exploración recibidos durante el experimento.

Estas métricas permiten analizar el balance entre exploración y explotación, así como la capacidad del agente para adaptarse a las distintas dinámicas markovianas.

9.5. Funciones de Experimentación

Para evaluar un agente, los estudiantes pueden utilizar:

```
# Evaluación simple de un agente para un problema específico
results = evaluate_problem1_agent(
    fixed_turns=True,           # True para horizonte fijo, False para
aleatorio
    exploration_bonus=0.1,      # Coeficiente β para el bono de
exploración
    n_experiments=100           # Número de experimentos a ejecutar
)

# Comparación con distintos valores de bono de exploración
results_varying_bonus = evaluate_problem1_with_varying_bonus(
    bonus_values=[0.0, 0.1, 0.5, 1.0],  # Valores de β a probar
    fixed_turns=True,                   # Horizonte fijo o aleatorio
    n_experiments=50                    # Experimentos por valor de β
)

# Evaluación de todos los agentes
all_results = run_all_agents(
    fixed_turns=True,
    exploration_bonus=0.1,
    n_experiments=100
)

# Comparación de todos los agentes con distintos valores de bono
all_bonus_results = run_all_agents_with_varying_bonus(
    bonus_values=[0.0, 0.1, 0.5, 1.0],
    fixed_turns=True,
    n_experiments=50
)
```

9.6. Visualizaciones

PROF

El framework proporciona visualizaciones automáticas que incluyen:

1. **Tablas de resumen estadístico** con media, desviación estándar, mínimo y máximo de cada métrica.
2. **Gráficos de barras** para recompensa media, porcentaje de acciones óptimas, arrepentimiento y bono de exploración.
3. **Gráficos comparativos** entre distintos valores de bono de exploración.
4. **Visualizaciones de la evolución** del rendimiento en función del bono de exploración.

9.7. Guía de Implementación

Para implementar agentes efectivos, los estudiantes deben tener en cuenta los siguientes aspectos:

9.7.1. Para el Problema 1:

- Mantener una **creencia** sobre el estado actual de cada brazo
- Actualizar esta creencia usando el **filtro forward** (ver Sección 7.2.1)
- Calcular el valor esperado de jalar cada brazo dada la creencia actual
- Balancear la explotación (elegir el brazo con mayor valor esperado) con la exploración (probar brazos menos explorados)

9.7.2. Para el Problema 2:

- Además de la creencia sobre el estado, mantener estimaciones de las matrices de transición
- Considerar un enfoque bayesiano o un método como Baum-Welch (Sección 7.2.3)
- La exploración es aún más importante aquí para refinar el conocimiento sobre las transiciones

9.7.3. Para el Problema 3:

- Mantener una creencia sobre el **estado conjunto**
- Tener en cuenta que jalar un brazo proporciona información sobre ambos
- Actualizar la creencia usando la **matriz de transición conjunta**

9.7.4. Para el Problema 4:

- Este es el caso más general y desafiante
- Implementar técnicas que puedan adaptarse a horizontes variables
- Considerar enfoques como **Thompson Sampling** o **UCB adaptado** (Sección 7.3)

9.8. Recomendaciones Adicionales

1. **Implementación incremental**: Comenzar con el Problema 1 y progresivamente adaptar la solución a los problemas más complejos.
2. **Experimentación con el bono de exploración**: Probar diferentes valores para el parámetro β y analizar cómo afecta el rendimiento.
3. **Verificación de actualización de creencias**: Asegurarse de que las creencias sobre los estados se actualizan correctamente después de cada acción y observación.
4. **Análisis de resultados**: Interpretar las métricas en contexto. Un alto porcentaje de acciones óptimas no siempre implica el mejor rendimiento a largo plazo si no se explora lo suficiente.
5. **Comparación de enfoques**: Implementar y comparar diferentes estrategias para el mismo problema (por ejemplo, ε-greedy vs. UCB vs. Thompson Sampling).

PROF

Los ejercicios buscan aplicar los conceptos teóricos de la toma de decisiones secuenciales en entornos markovianos parcialmente observables. La implementación exitosa requiere combinar los algoritmos de inferencia (Sección 7.2) con los algoritmos de decisión (Sección 7.3) para crear agentes que balanceen efectivamente la exploración y la explotación.