Cadenas de Markov

Investigación Operativa, Universidad de San Andrés

Si encuentran algún error en el documento o hay alguna duda, mandenmé un mail a rodriguezf@udesa.edu.ar y lo revisamos.

1. Introducción a las Cadenas de Markov

Una cadena de Markov es un proceso estocástico que cumple con la propiedad de Markov, es decir, la probabilidad de cualquier estado futuro depende únicamente del estado presente y no de la secuencia de eventos que le precedieron.

1.1. Definición Formal

Sea $\{X_n, n \geq 0\}$ un proceso estocástico con espacio de estados S. Decimos que es una cadena de Markov si para todo $n \geq 0$ y para todos los estados $i_0, i_1, ..., i_n, j \in S$:

$$P(X_{n+1} = j | X_n = i_n, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = P(X_{n+1} = j | X_n = i_n)$$

1.2. Matriz de Transición

La matriz de transición $P = (p_{ij})$ contiene las probabilidades de transición de un estado a otro, donde:

$$p_{ij} = P(X_{n+1} = j | X_n = i)$$

Propiedades importantes:

- $0 \leq p_{ij} \leq 1$ para todo $i, j \in S$
- $\sum_{j \in S} p_{ij} = 1$ para todo $i \in S$ (cada fila suma 1)

1.3. Ejemplo Introductorio

Consideremos un sistema muy simple con solo dos estados: Funcionando (F) y Roto (R).

- Si el sistema está funcionando hoy, hay 80 % de probabilidad de que siga funcionando mañana y 20 % de que se rompa.
- Si el sistema está roto hoy, hay 60 % de probabilidad de que sea reparado (vuelva a F) y 40 % de que siga roto.

La matriz de transición es:

$$P = \begin{pmatrix} 0{,}8 & 0{,}2 \\ 0{,}6 & 0{,}4 \end{pmatrix}$$

donde las filas representan el estado actual (F, R) y las columnas el estado futuro.

1.3.1. ¿Qué pasa en 2 días?

Si hoy el sistema está funcionando, ¿cuál es la probabilidad de que esté funcionando en 2 días? Debemos considerar todos los caminos posibles:

- F $\to$ F $\to$ F: $(0{,}8)(0{,}8) = 0{,}64$
- F $\to$ R $\to$ F: $(0{,}2)(0{,}6) = 0{,}12$

Total: $0{,}64 + 0{,}12 = 0{,}76$

Esto es equivalente a calcular P^2 (multiplicar la matriz por sí misma):

$$P^2 = \begin{pmatrix} 0{,}8 & 0{,}2 \\ 0{,}6 & 0{,}4 \end{pmatrix} \begin{pmatrix} 0{,}8 & 0{,}2 \\ 0{,}6 & 0{,}4 \end{pmatrix} = \begin{pmatrix} 0{,}76 & 0{,}24 \\ 0{,}72 & 0{,}28 \end{pmatrix}$$

El elemento $P^2_{FF} = 0{,}76$ confirma nuestro cálculo manual.

2. Ejercicios Básicos

2.1. Ejercicio 1: Predicción del Clima

Un meteorólogo estudia el clima en Buenos Aires y clasifica los días en tres estados posibles: Soleado (S), Nublado (N) y Lluvioso (L). Las probabilidades de transición son:

- Si hoy está soleado:
  - 70 % de probabilidad de que mañana esté soleado
  - 20 % de probabilidad de que esté nublado
  - 10 % de probabilidad de que llueva
- Si hoy está nublado:
  - 30 % de probabilidad de que mañana esté soleado
  - 40 % de probabilidad de que siga nublado
  - 30 % de probabilidad de que llueva
- Si hoy llueve:

- 20 % de probabilidad de que mañana esté soleado
- 40 % de probabilidad de que esté nublado
- 40 % de probabilidad de que siga lloviendo

a) Construir la matriz de transición P que representa este sistema.

b) ¿Cuál es la probabilidad de que llueva dentro de dos días si hoy está soleado?

Solución:

a) La matriz de transición P es:

$$P = \begin{pmatrix} 0{,}7 & 0{,}2 & 0{,}1 \\ 0{,}3 & 0{,}4 & 0{,}3 \\ 0{,}2 & 0{,}4 & 0{,}4 \end{pmatrix}$$

b) Si hoy está soleado, la probabilidad de que dentro de dos días llueva es:

$$P(X_2 = L|X_0 = S) = \sum_{j \in S} P(X_1 = j|X_0 = S)P(X_2 = L|X_1 = j)$$

Para calcular la probabilidad de que llueva en dos días partiendo de un día soleado, necesitamos considerar todos los caminos posibles:

- Si mañana está soleado (0.7) y luego llueve (0.1)
- Si mañana está nublado (0.2) y luego llueve (0.3)
- Si mañana llueve (0.1) y sigue lloviendo (0.4)

$$\begin{aligned} P(X_2 = L|X_0 = S) &= (0{,}7)(0{,}1) + (0{,}2)(0{,}3) + (0{,}1)(0{,}4) \\ &= 0{,}07 + 0{,}06 + 0{,}04 \\ &= 0{,}17 \end{aligned}$$

Por lo tanto, hay un 17 % de probabilidad de que llueva dentro de dos días si hoy está soleado.

2.2. Ejercicio 2: Puntos en Tenis

Final de Roland Garros. Francisco Cerúndolo está contra Holger Rune con match point. Siendo el analista de tenis de Cerúndolo, se sabe las siguientes probabilidades:

- Hay un 50 % de chances de ganar el punto de un ace o un saque no devuelto

- Tiene 30 % de chances de que la pelota entre con el primer saque y se arme el punto
- Si erra el primer saque, tiene 90 % de chances de meter el segundo
- Hay un 2 % de chances de que gane el punto directamente con el segundo saque
- En cualquier peloteo (sea con primer o segundo saque), tiene 55 % de chances de ganar el punto

Resuelva los siguientes items:

a) Identifique los estados posibles del sistema y construya el diagrama de transición.

b) Calcule la probabilidad de que Cerúndolo gane el punto y se consagre campeón.

c) Si se jugaran infinitos puntos con estas probabilidades, ¿qué porcentaje ganaría cada jugador?

Solución:

a) Los estados posibles son:

- S: Primer saque de Cerúndolo
- D: Segundo saque de Cerúndolo
- P: Punto en juego (peloteo)
- W: Punto ganado por Cerúndolo
- L: Punto perdido por Cerúndolo (ganado por Rune)

b) La matriz de transición es:

$$P = \begin{pmatrix} 0 & 0{,}20 & 0{,}30 & 0{,}50 & 0 \\ 0 & 0 & 0{,}90 & 0{,}02 & 0{,}08 \\ 0 & 0 & 0 & 0{,}55 & 0{,}45 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Donde las filas y columnas siguen el orden S, D, P, W, L.

Analicemos cada fila:

- Primera fila (S): 0.20 de ir a D (errar primer saque), 0.30 de ir a P (peloteo), 0.50 de ir a W (ganar directo)

- Segunda fila (D): 0.90 de ir a P (peloteo), 0.02 de ir a W (ganar directo), 0.08 de ir a L (doble falta)
- Tercera fila (P): 0.55 de ir a W (ganar el punto), 0.45 de ir a L (perder el punto)
- Cuarta y quinta filas (W y L): estados absorbentes (probabilidad 1 de quedarse en el mismo estado)

La probabilidad de que Cerúndolo gane el punto se puede calcular considerando todos los caminos posibles:

- Ganar directo con el primer saque (0.50)
- Ir a peloteo con primer saque (0.30) y ganar el peloteo (0.55)
- Errar primer saque (0.20), meter segundo (0.90) y ganar el peloteo (0.55)
- Errar primer saque (0.20), ganar directo con el segundo saque (0.02)

Resolvemos entonces la ecuación:

$$\begin{aligned} P(\text{Cerúndolo gana}) &= 0{,}50 + (0{,}30 \cdot 0{,}55) + (0{,}20 \cdot 0{,}90 \cdot 0{,}55) + (0{,}20 \cdot 0{,}02) \\ &= 0{,}50 + 0{,}165 + 0{,}099 + 0{,}004 \\ &= 0{,}768 \end{aligned}$$

Por lo tanto, Cerúndolo tiene aproximadamente un 76.8 % de probabilidad de ganar el punto.

c) A largo plazo, considerando que cada punto es independiente y las probabilidades se mantienen constantes:

- Cerúndolo ganaría aproximadamente el 76.8 % de los puntos
- Rune ganaría aproximadamente el 23.2 % de los puntos

2.3. Ejercicio 3: Máquina de Juguetes

En una fábrica de juguetes, hay una máquina que produce muñecos de peluche. La máquina puede estar en tres estados:

- F: Funcionando perfectamente (0.80)
- M: Mal funcionamiento (0.15)
- R: Rota (0.05)

Cuando la máquina está funcionando perfectamente, produce 100 muñecos por hora. En mal funcionamiento, produce 50 muñecos por hora. Cuando está rota, no produce nada.

El técnico de mantenimiento puede:

- Reparar la máquina (0.90)
- Empeorar la situación (0.10)

Resuelva los siguientes items:

a) Construya la matriz de transición considerando los estados de la máquina y las acciones del técnico.

b) Si la máquina comienza funcionando perfectamente, ¿cuál es la probabilidad de que esté rota después de dos intervenciones del técnico?

c) ¿Cuál es la probabilidad de que la máquina funcione perfectamente durante tres horas consecutivas?

Ayuda:

- La máquina puede cambiar de estado por sí sola o por la intervención del técnico
- El técnico solo interviene cuando la máquina no está funcionando perfectamente
- Las probabilidades son independientes del tiempo

Resolución:

a) La matriz de transición es:

$$P = \begin{pmatrix} 0{,}80 & 0{,}15 & 0{,}05 \\ 0{,}90 & 0 & 0{,}10 \\ 0{,}90 & 0 & 0{,}10 \end{pmatrix}$$

Donde las filas y columnas siguen el orden F, M, R.

Explicación de los valores:

- Primera fila: Desde F, la máquina tiene 80 % de probabilidad de seguir en F, 15 % de pasar a M y 5 % de pasar a R
- Segunda fila: Desde M, el técnico tiene 90 % de probabilidad de arreglarla (F), 0 % de dejarla igual (M) y 10 % de empeorarla (R)
- Tercera fila: Desde R, el técnico tiene 90 % de probabilidad de arreglarla (F), 0 % de dejarla en M y 10 % de dejarla igual (R)

b) Para calcular la probabilidad de que esté rota después de dos intervenciones, partiendo de F, analizamos todas las posibles combinaciones válidas:

Posibles caminos, sin considerar que la arregla porque si no sería infinito:

- F → M → M → R
- F → R → R → R
- F → R → M → R
- F → M → R → R

Calculando cada probabilidad:

$$
\begin{aligned}
P(F \to M \to M \to R) &= 0{,}15 \cdot 0 \cdot 0{,}10 = 0 \\
P(F \to R \to R \to R) &= 0{,}05 \cdot 0{,}10 \cdot 0{,}10 = 0{,}0005 \\
P(F \to R \to M \to R) &= 0{,}05 \cdot 0 \cdot 0{,}10 = 0 \\
P(F \to M \to R \to R) &= 0{,}15 \cdot 0{,}10 \cdot 0{,}10 = 0{,}0015
\end{aligned}
$$

Sumando todas las probabilidades:

$$
\begin{aligned}
P(\text{Rota después de 2 intervenciones}) &= 0 + 0{,}0005 + 0 + 0{,}0015 \\
&= 0{,}002
\end{aligned}
$$

c) Para que funcione perfectamente durante tres horas consecutivas:

$$
\begin{aligned}
P(\text{F durante 3 horas}) &= 0{,}80 \cdot 0{,}80 \cdot 0{,}80 \\
&= 0{,}512
\end{aligned}
$$

3. Clasificación de Cadenas de Markov

3.1. Estados Comunicantes

Decimos que el estado i **comunica** con el estado j (denotado $i \to j$) si existe algún número $n \geq 0$ tal que $P_{ij}^{(n)} > 0$, es decir, si es posible llegar del estado i al estado j en un número finito de pasos.

Dos estados i y j se **intercomunican** si $i \to j$ y $j \to i$.

3.2. Cadenas Irreducibles

Una cadena de Markov es **irreducible** si todos sus estados se intercomunican, es decir, desde cualquier estado es posible llegar a cualquier otro estado en un número finito de pasos.

Ejemplo: La cadena del clima (Ejercicio 1) es irreducible porque desde cualquier estado (Soleado, Nublado, Lluvioso) se puede llegar a cualquier otro estado.

3.3. Periodicidad

El **período** de un estado i es el máximo común divisor de todos los valores de n para los cuales $P_{ii}^{(n)} > 0$.

Un estado con período 1 se llama **aperiódico**. Una cadena es aperiódica si todos sus estados son aperiódicos.

Intuición: Un estado periódico solo puede volver a sí mismo en múltiplos de su período. Por ejemplo, si el período es 2, solo puede volver en 2, 4, 6, etc. pasos.

3.4. Estados Absorbentes

Un estado i es **absorbente** si $p_{ii} = 1$, es decir, una vez que el sistema entra en ese estado, nunca puede salir de él.

Ejemplo: En el Ejercicio 2 (Tenis), los estados W (Cerúndolo gana) y L (Cerúndolo pierde) son estados absorbentes, porque una vez terminado el punto, no hay transiciones a otros estados.

3.5. Cadenas Ergódicas

Una cadena de Markov es **ergódica** si es **irreducible** y **aperiódica**. Las cadenas ergódicas tienen una propiedad importante: poseen una única distribución estacionaria π hacia la cual converge el sistema independientemente del estado inicial.

4. Conceptos Avanzados

4.1. Ecuaciones de Chapman-Kolmogorov

Intuición: Si queremos saber la probabilidad de ir del estado i al estado j en $n + m$ pasos, podemos pensarlo como: primero damos n pasos (llegamos a algún estado intermedio k), y luego desde k damos m pasos más hasta llegar a j.

Para calcular las probabilidades de transición en n pasos, definimos:

$$P_{ij}^{(n)} = P(X_n = j | X_0 = i)$$

Se cumple la ecuación de Chapman-Kolmogorov:

$$P_{ij}^{(n+m)} = \sum_{k \in S} P_{ik}^{(n)} \cdot P_{kj}^{(m)}$$

En forma matricial:

$$P^{(n+m)} = P^{(n)} \cdot P^{(m)}, \quad P^{(n)} = P^n$$

Esto significa que para calcular probabilidades de transición a n pasos, simplemente elevamos la matriz de transición P a la potencia n.

Conexión con el Ejercicio 1: Cuando calculamos la probabilidad de lluvia en 2 días partiendo de soleado, estuvimos usando implícitamente Chapman-Kolmogorov: sumamos sobre todos los estados intermedios posibles (soleado, nublado, lluvioso) en el día 1.

4.2. Distribución en el Tiempo n

Intuición: Si comenzamos con una distribución de probabilidad inicial $q^{(0)}$ (cuánto porcentaje en cada estado al tiempo 0), podemos calcular cómo se distribuye el sistema en cualquier tiempo futuro n.

Sea $q^{(0)}$ el vector fila inicial con $q_i^{(0)} = P(X_0 = i)$. La distribución de probabilidad en el tiempo n está dada por:

$$q^{(n)} = q^{(0)} \cdot P^n$$

donde $q_j^{(n)} = P(X_n = j)$. Esta fórmula permite predecir la distribución futura del sistema conociendo su distribución inicial.

Ejemplo: Si al inicio el clima tiene 50 % de probabilidad de estar soleado, 30 % nublado y 20 % lluvioso, entonces $q^{(0)} = [0{,}5, 0{,}3, 0{,}2]$. Para saber la distribución en 5 días, calculamos $q^{(5)} = q^{(0)} \cdot P^5$.

4.3. Estado Estacionario

Motivación: ¿Qué pasa si aplicamos la matriz de transición P infinitas veces? ¿Converge la distribución del sistema a algo fijo?

En cadenas de Markov **ergódicas** (irreducibles y aperiódicas) existe un vector de probabilidades estacionarias π tal que:

$$\pi = \pi \cdot P, \quad \sum_{j \in S} \pi_j = 1$$

¿Por qué esta ecuación? Si π representa una distribución estacionaria, significa que al aplicar la matriz de transición P, la distribución no cambia: sigue siendo π. Es decir, el sistema está en equilibrio.

Para encontrar π, resolvemos el sistema de ecuaciones:

$$\pi_j = \sum_{i \in S} \pi_i \cdot p_{ij}, \quad \sum_{j \in S} \pi_j = 1$$

4.3.1. Método práctico para resolver el sistema

Para resolver el sistema podemos seguir los siguientes pasos:

1. Escribir la ecuación $\pi = \pi \cdot P$ como un sistema de ecuaciones lineales.
2. La ecuación es equivalente a $(\pi \cdot P)^T = \pi^T$, o sea $(P^T - I)\pi^T = 0$.
3. Este sistema tiene infinitas soluciones (sistema homogéneo). Una de las ecuaciones es redundante.
4. Reemplazar una ecuación por $\sum_j \pi_j = 1$ (condición de normalización).
5. Resolver el sistema resultante usando Python.

El valor π_j representa la fracción de tiempo que el sistema pasa en el estado j a largo plazo.

4.4. Pasos Prácticos de Cálculo

Para resolver problemas con cadenas de Markov:

1. Definir el conjunto de estados S y construir la matriz de transición P.
2. Clasificar la cadena (irreducible, ergódica, con estados absorbentes, etc.).
3. Calcular P^n para probabilidades a n pasos, o usar $q^{(n)} = q^{(0)} \cdot P^n$.
4. Si la cadena es ergódica, resolver $\pi = \pi \cdot P$ para obtener el estado estacionario.
5. Interpretar los resultados en el contexto del problema.

4.5. Importancia en Investigación Operativa

Las cadenas de Markov son fundamentales en Investigación Operativa porque aplican para temas que vemos mas adelante en la materia:

- Modelan sistemas de colas, inventarios, fiabilidad y comportamiento de clientes.
- Permiten predecir el desempeño del sistema y diseñar políticas óptimas.
- Cuantifican tiempos de espera y probabilidades de eventos críticos.

5. Ejercicios Intermedios

5.1. Ejercicio 4: Demanda en Centro de Datos

Un datacenter de una empresa de cloud computing puede tener tres niveles de demanda diaria:

- Baja demanda (B),
- Media demanda (M),
- Alta demanda (A).

Según los registros históricos:

- Si está en baja demanda:
  - 0.3 de permanecer en baja,
  - 0.5 de pasar a media,
  - 0.2 de pasar a alta.
- Si está en media demanda:
  - 0.2 de pasar a baja,
  - 0.4 de permanecer en media,
  - 0.4 de pasar a alta.
- Si está en alta demanda:
  - 0.3 de pasar a baja,
  - 0.5 de pasar a media,
  - 0.2 de permanecer en alta.

Se modela como una cadena de Markov. Se desea conocer la probabilidad de que en 3 días el sistema esté en demanda alta, si hoy se encuentra en demanda baja.

<u>Solución:</u>

5.1.1. Estados:

Denotamos los estados como:

$$\text{Baja } (0), \quad \text{Media } (1), \quad \text{Alta } (2)$$

5.1.2. Matriz de transición:

$$P = \begin{pmatrix} 0{,}3 & 0{,}5 & 0{,}2 \\ 0{,}2 & 0{,}4 & 0{,}4 \\ 0{,}3 & 0{,}5 & 0{,}2 \end{pmatrix}$$

Cada fila representa el estado actual y cada columna el estado futuro.

5.1.3. Estado inicial:

Como el sistema comienza en demanda baja (estado 0), el vector inicial es:

$$\mathbf{e_0} = [1 \quad 0 \quad 0]$$

5.1.4. Cálculo de P^3:

Calculamos la matriz de transición a 3 pasos:

$$\mathbf{e_0} \cdot P^3 = \text{probabilidades de estar en cada estado al día 3}$$

5.1.5. Resultado (usando Python):

Mediante multiplicación matricial, se obtiene:

$$\mathbf{e_0} \cdot P^3 = [0{,}248, \quad 0{,}452, \quad \boxed{0{,}300}]$$

5.1.6. Conclusión:

La probabilidad de que el sistema esté en **demanda alta** en 3 días, comenzando en baja demanda, es:

$$\boxed{0{,}300}$$

5.2. Ejercicio 5: Movilidad del Ingeniero Civil

Un ingeniero civil trabaja en tres ciudades: A, B y C. Cada día permanece en la ciudad o se desplaza a otra, dependiendo de la demanda de trabajo. Las probabilidades de transición entre ciudades son:

- Si trabaja en C:

$$P(C \to C) = 0{,}4, \quad P(C \to B) = 0{,}4, \quad P(C \to A) = 0{,}2$$

- Si trabaja en B:

$$P(B \to B) = 0{,}2, \quad P(B \to C) = 0{,}6, \quad P(B \to A) = 0{,}2$$

- Si trabaja en A:

$$P(A \to A) = 0{,}1, \quad P(A \to B) = 0{,}3, \quad P(A \to C) = 0{,}6$$

a) ¿Cuál es la probabilidad de que esté en la ciudad C después de 2 días, si hoy está en C?

b) ¿Cuál es la distribución estacionaria a largo plazo?

<u>Solución:</u>

5.2.1. Definición de estados:

Codificamos las ciudades como:

$$A = 0, \quad B = 1, \quad C = 2$$

5.2.2. Matriz de transición P:

$$P = \begin{pmatrix} 0{,}1 & 0{,}3 & 0{,}6 \\ 0{,}2 & 0{,}2 & 0{,}6 \\ 0{,}2 & 0{,}4 & 0{,}4 \end{pmatrix}$$

(Las filas corresponden al estado actual, las columnas al estado siguiente.)

5.2.3. Parte a) Probabilidad de estar en C al cabo de 2 días (partiendo de C):

$$\text{Estado inicial: } \mathbf{e_2} = [0 \quad 0 \quad 1]$$
$$\text{Buscamos: } \mathbf{e_2} \cdot P^2$$

5.2.4. Resultado (cálculo con Python):

$$\mathbf{e_2} \cdot P^2 = [0{,}26 \quad 0{,}34 \quad \boxed{0{,}40}]$$

La probabilidad de estar en la ciudad C después de 2 días, comenzando en C, es:

$$\boxed{0{,}40}$$

5.2.5. Parte b) Distribución estacionaria (a largo plazo):

Buscamos el vector $\pi = [\pi_A \quad \pi_B \quad \pi_C]$ tal que:

$$\pi \cdot P = \pi, \quad \pi_A + \pi_B + \pi_C = 1$$

Resolviendo el sistema lineal, se obtiene:

$$\pi = \left[\frac{12}{71}, \quad \frac{22}{71}, \quad \frac{37}{71}\right]$$

En el largo plazo, el ingeniero pasa el:

$$\boxed{\frac{12}{71} \approx 16{,}9\,\% \text{ en A}}, \quad \boxed{\frac{22}{71} \approx 31{,}0\,\% \text{ en B}}, \quad \boxed{\frac{37}{71} \approx 52{,}1\,\% \text{ en C}}$$

6. Procesos de Decisión de Markov (MDP)

6.1. Motivación: De Cadenas Pasivas a Decisiones Activas

Hasta ahora vimos cadenas de Markov donde el sistema evoluciona según probabilidades fijas. No podemos influir en el comportamiento del sistema: simplemente observamos y calculamos probabilidades.

En muchas situaciones reales, podemos **tomar decisiones** que afectan cómo evoluciona el sistema:

- ¿Cuándo reparar una máquina?
- ¿Cuándo reemplazarla por una nueva?
- ¿Dónde estacionar el auto para minimizar costos?
- ¿Cuándo hacer mantenimiento preventivo?

Los **Procesos de Decisión de Markov (MDP)** extienden las cadenas de Markov:

1. **Decisiones**: En cada estado podemos elegir entre varias acciones.
2. **Transiciones dependientes de decisiones**: La probabilidad de ir al siguiente estado depende de la acción elegida.
3. **Costos o recompensas**: Cada decisión tiene un costo (o beneficio) asociado.

Objetivo: Encontrar la *política óptima* (qué decisión tomar en cada estado) que minimice el costo promedio a largo plazo (o maximice la recompensa).

6.2. Formulación General

En un MDP, para cada estado $i = 0, \ldots, M$ y decisión $k = 1, \ldots, K$, definimos:

- $p_{ij}(k)$: probabilidad de transición del estado i al estado j bajo la decisión k.
- c_{ik}: costo asociado a tomar la decisión k en el estado i.
- y_{ik}: variable de decisión que representa $P(\text{estado} = i \text{ y decisión } = k)$.

La variable y_{ik} puede interpretarse como:

$$y_{ik} = \pi_i \cdot D_{ik}$$

donde π_i es la probabilidad estacionaria de estar en el estado i y D_{ik} es la probabilidad de tomar la decisión k dado que se está en el estado i.

6.3. Formulación por Programación Lineal

El objetivo es minimizar el costo promedio esperado a largo plazo:

$$\text{Minimizar} \quad E(C) = \sum_{i=0}^{M} \sum_{k=1}^{K} c_{ik} \cdot y_{ik}$$

sujeto a las siguientes restricciones:

6.3.1. Normalización:

$$\sum_{i=0}^{M} \sum_{k=1}^{K} y_{ik} = 1$$

Esto garantiza que las y_{ik} formen una distribución de probabilidad válida.

6.3.2. Balance de flujo en cada estado j:

$$\sum_{k=1}^{K} y_{jk} - \sum_{i=0}^{M} \sum_{k=1}^{K} y_{ik} \cdot p_{ij}(k) = 0$$

Intuición del balance de flujo: Imaginemos que el sistema es una red de estados por la cual "fluye" probabilidad. Para cada estado j:

- **Lado izquierdo** $\sum_k y_{jk}$: frecuencia con que se está en estado j y se toma alguna acción (flujo que sale).

- **Lado derecho** $\sum_{i,k} y_{ik} \cdot p_{ij}(k)$: frecuencia con que se llega al estado j desde otros estados i tomando decisiones k (flujo que entra).

La ecuación dice: *lo que sale del estado j debe ser igual a lo que entra al estado j*. Esto garantiza que el sistema esté en equilibrio estacionario.

Considere el estado j. Las flechas que entran representan flujos desde estados i con decisiones k que tienen probabilidad $p_{ij}(k)$ de llegar a j. Las flechas que salen representan las decisiones tomadas en j. El balance requiere que la suma de flujos entrantes iguale la suma de flujos salientes.

6.3.3. No negatividad:

$$y_{ik} \geq 0 \quad \forall\, i = 0, \ldots, M,\ k = 1, \ldots, K$$

Una vez resuelto el problema de programación lineal, la probabilidad estacionaria π_i y la política óptima se obtienen por:

$$\pi_i = \sum_{k=1}^{K} y_{ik}$$

$$\text{Política óptima en estado } i: \quad D_{ik} = \frac{y_{ik}}{\pi_i}$$

6.3.4. Interpretación de las Variables de Decisión

Las variables y_{ik} pueden tomar valores reales entre 0 y 1, lo que permite políticas probabilísticas:

- Si $y_{ik} = 0$: nunca se toma la decisión k en el estado i.
- Si $y_{ik} > 0$: existe una probabilidad de tomar la decisión k en el estado i.
- La suma $\sum_k D_{ik} = 1$ para cada estado i, garantizando que se tome alguna decisión.

Esta formulación convierte un problema de decisión secuencial en un problema de programación lineal que puede resolverse como ya sabemos.

7. Ejercicios de MDP

7.1. Ejercicio 6: Reemplazo de Maquinaria (MDP)

Una empresa utiliza maquinaria que se desgasta con el tiempo. Cada máquina puede estar en uno de cuatro estados:

- Estado 0: máquina nueva.
- Estado 1: máquina con poco desgaste.
- Estado 2: máquina deteriorada.
- Estado 3: máquina rota.

En cada período, la empresa puede tomar una de las siguientes decisiones:

1. No hacer nada ($k = 1$).
2. Renovar la máquina ($k = 2$, solo aplicable en estado 2).
3. Comprar una máquina nueva ($k = 3$, aplicable en estados 1, 2 o 3).

El objetivo es encontrar la política óptima (probabilística) que minimice el costo promedio a largo plazo.

Matriz de transición sin intervención:

Cuando no se toma ninguna acción, la máquina evoluciona según:

Estado $i \to j$	0	1	2	3
0	0	7/8	1/16	1/16
1	0	3/4	1/8	1/8
2	0	0	1/2	1/2
3	0	0	0	1

Efecto de las decisiones:

- **Renovar** en el estado 2 (acción $k = 2$) lleva con certeza al estado 1.
- **Comprar nueva** en los estados 1, 2 o 3 (acción $k = 3$) lleva con certeza al estado 0.

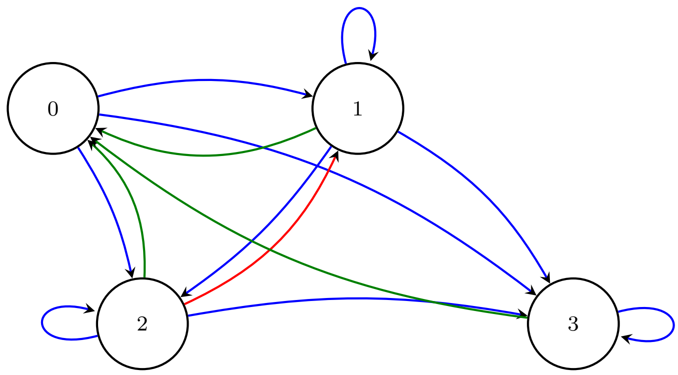
Costos de las decisiones (en miles de dólares):

Estado i	c_{ik} Nada ($k = 1$)	Renovar ($k = 2$)	Nueva ($k = 3$)
0	0	—	—
1	1	—	6
2	3	4	6
3	—	—	6

Solución:

7.1.1. Diagrama de la Cadena de Markov con Decisiones:

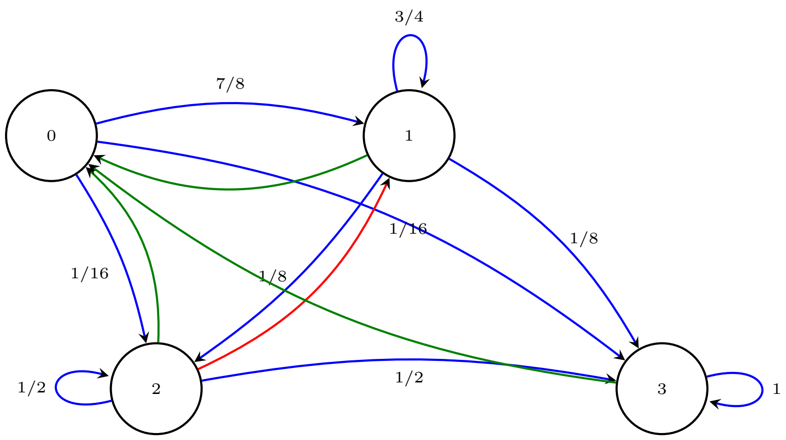
El siguiente diagrama muestra visualmente todas las transiciones posibles y las decisiones que se pueden tomar en cada estado:



Interpretación del diagrama:

- Flechas azules: Decisión 1 (no hacer nada) - el sistema evoluciona naturalmente según la matriz de transición.
- Flecha roja: Decisión 2 (renovar) - solo disponible en estado 2, lleva al estado 1.
- Flechas verdes: Decisión 3 (comprar nueva) - disponible en estados 1, 2, 3, siempre lleva al estado 0.

El siguiente diagrama muestra las mismas transiciones pero con las probabilidades específicas en las transiciones azules (evolución natural del sistema):



Nota: Las flechas rojas y verdes representan decisiones determinísticas (renovar o comprar nueva), por lo que siempre llevan al estado destino con certeza. Solo las flechas azules muestran probabilidades porque representan la evolución natural del sistema cuando no se interviene.

Las variables válidas son: $y_{01}, y_{11}, y_{13}, y_{21}, y_{22}, y_{23}, y_{33}$. Cada variable y_{ik} representa la probabilidad de estar en el estado i y tomar la decisión k.

7.1.2. Función objetivo:

La función objetivo la podemos plantear de la siguiente manera:

$$\text{Minimizar} \quad Z = 1000\, y_{11} + 6000\, y_{13} + 3000\, y_{21} + 4000\, y_{22} + 6000\, y_{23} + 6000\, y_{33}$$

7.1.3. Restricción de normalización:

La restricción de normalización la podemos plantear de la siguiente manera:

$$y_{01} + y_{11} + y_{13} + y_{21} + y_{22} + y_{23} + y_{33} = 1$$

7.1.4. Restricciones de balance de flujo:

Para cada estado j, el flujo que sale debe ser igual al flujo que entra.

Estado 0 (máquina nueva):

La única forma de salir del estado 0 es mediante y_{01} (no hacer nada). Se puede llegar al estado 0 comprando una máquina nueva desde los estados 1, 2 o 3:

$$y_{01} = y_{13} + y_{23} + y_{33}$$

Estado 1 (poco desgaste):

Se sale del estado 1 mediante y_{11} (no hacer nada) o y_{13} (comprar nueva). Se puede llegar al estado 1 desde:

- Estado 0 con probabilidad $\frac{7}{8}$ (acción $k = 1$)
- Estado 1 con probabilidad $\frac{3}{4}$ (acción $k = 1$)
- Estado 2 con probabilidad 1 (acción $k = 2$, renovar)

$$y_{11} + y_{13} = \frac{7}{8} y_{01} + \frac{3}{4} y_{11} + y_{22}$$

Estado 2 (deteriorada):

Se sale del estado 2 mediante y_{21}, y_{22} o y_{23}. Se puede llegar al estado 2 desde:

- Estado 0 con probabilidad $\frac{1}{16}$ (acción $k = 1$)
- Estado 1 con probabilidad $\frac{1}{8}$ (acción $k = 1$)
- Estado 2 con probabilidad $\frac{1}{2}$ (acción $k = 1$)

$$y_{21} + y_{22} + y_{23} = \frac{1}{16}\, y_{01} + \frac{1}{8}\, y_{11} + \frac{1}{2}\, y_{21}$$

Estado 3 (rota):

Se sale del estado 3 mediante y_{33} (comprar nueva). Se puede llegar al estado 3 desde:

- Estado 0 con probabilidad $\frac{1}{16}$ (acción $k = 1$)
- Estado 1 con probabilidad $\frac{1}{8}$ (acción $k = 1$)
- Estado 2 con probabilidad $\frac{1}{2}$ (acción $k = 1$)

$$y_{33} = \frac{1}{16}\, y_{01} + \frac{1}{8}\, y_{11} + \frac{1}{2}\, y_{21}$$

7.1.5. No negatividad:

$$y_{ik} \geq 0 \quad \forall\, i, k$$

7.1.6. Solución óptima:

Aplicando el método simplex, se obtiene:

$$y_{01} = \frac{2}{21}, \quad y_{11} = \frac{5}{7}, \quad y_{13} = 0, \quad y_{21} = 0, \quad y_{22} = \frac{2}{21}, \quad y_{23} = 0, \quad y_{33} = \frac{2}{21}$$

7.1.7. Política óptima:

La política óptima se obtiene calculando $D_{ik} = \frac{y_{ik}}{\pi_i}$, donde $\pi_i = \sum_k y_{ik}$:

- **Estado 0**: $D_{01} = 1$ (no hacer nada)
- **Estado 1**: $D_{11} = 1$, $D_{13} = 0$ (no hacer nada)
- **Estado 2**: $D_{21} = 0$, $D_{22} = 1$, $D_{23} = 0$ (renovar parcialmente)
- **Estado 3**: $D_{33} = 1$ (comprar nueva)

7.1.8. Costo promedio óptimo:

El costo promedio mínimo a largo plazo es:

$$Z^* = 1000 \cdot \frac{5}{7} + 6000 \cdot 0 + 3000 \cdot 0 + 4000 \cdot \frac{2}{21} + 6000 \cdot 0 + 6000 \cdot \frac{2}{21} \approx 1666{,}67 \text{ dólares/semana}$$

7.1.9. Interpretación de las restricciones de conservación de flujo:

Las restricciones de balance de flujo garantizan que el sistema esté en equilibrio estacionario. Por ejemplo, para el estado 1:

$$\underbrace{y_{11} + y_{13}}_{\text{flujo que sale}} = \underbrace{\frac{7}{8} y_{01} + \frac{3}{4} y_{11} + y_{22}}_{\text{flujo que entra}}$$

- **Lado izquierdo**: frecuencia con que se está en estado 1 y se actúa.
- **Lado derecho**: frecuencia con que se llega al estado 1 desde otros estados.
- La igualdad garantiza que el flujo total hacia y desde el estado 1 se equilibra.

Esta formulación convierte el problema de decisión secuencial en un problema de programación lineal que podemos resolver como ya sabemos.

7.1.10. Simulación en Python:

El siguiente código muestra cómo simular el comportamiento del sistema con y sin la política de reemplazo:

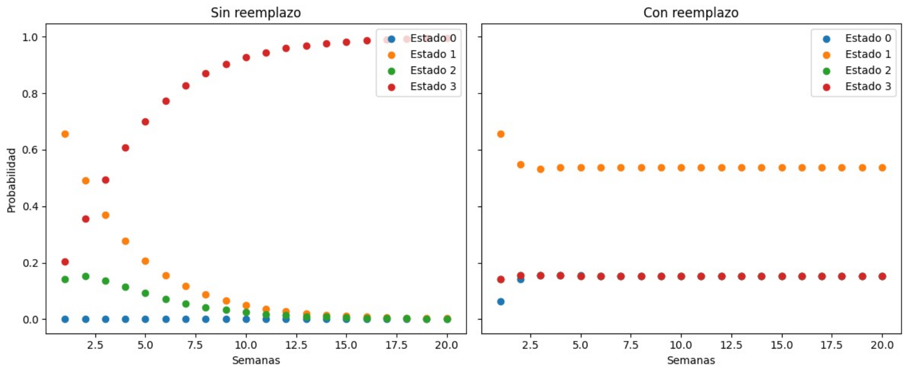
```
import numpy as np

# Matriz de transicion si NO se hace nada (accion "normal
   ")
# Estados: 0->1->2->3 (3 es absorbente)
P_normal = np.array([
    [0,   7/8, 1/16, 1/16],
    [0,   3/4, 1/8,  1/8 ],
    [0,   0,   1/2,  1/2 ],
    [0,   0,   0,    1   ]
])

# Matriz de transicion si REEMPLAZAMOS al llegar a estado
    3
P_reemplazo = np.array([
    [0,   7/8, 1/16, 1/16],
    [0,   3/4, 1/8,  1/8 ],
```

```
    [0,   0,   1/2,  1/2 ],
    [1,   0,   0,    0   ]  # Desde 3 volvemos a 0
])

# Numero de semanas a simular
semanas = 10

# Calculamos P^n por multiplicacion sucesiva
P_normal_n = P_normal.copy()
P_reemplazo_n = P_reemplazo.copy()

for i in range(1, semanas):
    P_normal_n = np.dot(P_normal_n, P_normal)
    P_reemplazo_n = np.dot(P_reemplazo_n, P_reemplazo)

# Vector estado inicial: comenzamos 100% en el estado 0
e_0 = np.array([1, 0, 0, 0])

# Distribucion tras 10 semanas
estado_sin_reemplazo = np.dot(e_0, P_normal_n)
estado_con_reemplazo = np.dot(e_0, P_reemplazo_n)

print(f"Distribucion tras {semanas} semanas (sin
    reemplazo):")
print(estado_sin_reemplazo)
# [0.    0.05  0.02  0.93]

print(f"\nDistribucion tras {semanas} semanas (con
    reemplazo):")
print(estado_con_reemplazo)
# [0.15  0.54  0.15  0.15]
```

7.1.11. Análisis de resultados:

- **Sin reemplazo:** Después de 10 semanas, hay un 93 % de probabilidad de que la máquina esté en estado 3 (rota). El sistema converge hacia el estado absorbente.
- **Con reemplazo:** El sistema alcanza un estado estacionario donde la máquina pasa aproximadamente 54 % del tiempo en estado 1, 15 % en estados 0 y 2, y 15 % en estado 3 (cuando es reemplazada inmediatamente).

La siguiente figura muestra la evolución de las probabilidades a lo largo del tiempo para ambas políticas:



Se observa claramente cómo sin reemplazo (izquierda), el sistema inevitablemente converge hacia el estado 3 (rota), mientras que con reemplazo (derecha), el sistema se estabiliza en una distribución estacionaria.

7.1.12. Implementacion completa con picos:

Para resolver el problema de optimizacion utilizando programacion lineal, podemos usar la biblioteca `picos`:

```
import picos as pc

# Crear el problema
prob = pc.Problem()

# Crear las variables validas individualmente
y_vars = {}  # y_vars[(estado, decision)]
valid = [(0, 0),          # y_01
         (1, 0), (1, 2),  # y_11, y_13
         (2, 0), (2, 1), (2, 2),  # y_21, y_22, y_23
         (3, 2)]          # y_33

for (i, k) in valid:
    y_vars[(i, k)] = pc.RealVariable(f"y{i}{k}", lower=0)

# Funcion objetivo
objective = (
    1000 * y_vars[(0, 0)] +
    6000 * y_vars[(1, 2)] +
    3000 * y_vars[(2, 0)] +
    4000 * y_vars[(2, 1)] +
    6000 * y_vars[(2, 2)] +
    6000 * y_vars[(3, 2)]
)
prob.set_objective("min", objective)

# Restriccion de suma total
```

```python
prob.add_constraint(sum(y_vars.values()) == 1)

# Restriccion 2: estado 0
prob.add_constraint(y_vars[(0, 0)] -
    (y_vars.get((1, 2), 0) + y_vars.get((2, 2), 0) +
     y_vars.get((3, 2), 0)) == 0)

# Restriccion 3: estado 1
prob.add_constraint(y_vars.get((1, 0), 0) +
    y_vars.get((1, 2), 0) -
    (7/8 * y_vars.get((0, 0), 0) +
     3/4 * y_vars.get((1, 0), 0) +
     y_vars.get((2, 1), 0)) == 0)

# Restriccion 4: estado 2
prob.add_constraint(y_vars.get((2, 0), 0) +
    y_vars.get((2, 1), 0) + y_vars.get((2, 2), 0) -
    (1/16 * y_vars.get((0, 0), 0) +
     1/8 * y_vars.get((1, 0), 0) +
     1/2 * y_vars.get((2, 0), 0)) == 0)

# Restriccion 5: estado 3
prob.add_constraint(y_vars.get((3, 2), 0) -
    (1/16 * y_vars.get((0, 0), 0) +
     1/8 * y_vars.get((1, 0), 0) +
     1/2 * y_vars.get((2, 0), 0)) == 0)

# Resolver
solution = prob.solve(solver='cvxopt')

# Mostrar resultados
print("Solucion optima:")
for (i, k), var in y_vars.items():
    print(f"y_{i}{k} = {var.value}")

print(f"\nCosto minimo: ${objective.value:.2f}/semana")
```

7.2. Ejercicio 7: Problema de Estacionamiento

Ramiro está preocupado por su Renault Coupé Fuego 1.6 Turbo 1985 y no le gustan las abolladuras. Cuando conduce a la Universidad de San Andrés, tiene las siguientes opciones de estacionamiento:

1. Estacionarlo en la calle ocupando un espacio.
2. Estacionarlo en la calle ocupando dos espacios.
3. Estacionarlo en el lote.

Además, si su auto se abolla, puede:

4. Repararlo (queda fuera de servicio por 1 día).
5. Conducir su auto abollado.

Probabilidades y costos asociados:

- Si estaciona en la calle en un espacio: probabilidad de abolladura = $\frac{1}{10}$, costo = \$0.
- Si estaciona en la calle en dos espacios: probabilidad de abolladura = $\frac{1}{50}$, probabilidad de multa = $\frac{3}{10}$, costo promedio = \$4,5 (multa de \$15).
- Si estaciona en el lote: probabilidad de abolladura = 0, costo = \$5.
- Si repara el auto: costo = \$50 (incluye tarifas de remis).
- Si conduce abollado: pérdida de valor y orgullo equivalente a = \$9 por día.

El objetivo es determinar la política óptima para minimizar el costo promedio esperado a largo plazo por día en la universidad.

Solución:

7.2.1. Estados del sistema:

- Estado 0: Auto sin abolladuras.
- Estado 1: Auto con abolladuras.

7.2.2. Acciones posibles:

- En estado 0:
  - Acción 1 ($k = 1$): Estacionar en 1 espacio.
  - Acción 2 ($k = 2$): Estacionar en 2 espacios.
  - Acción 3 ($k = 3$): Estacionar en lote.
- En estado 1:
  - Acción 4 ($k = 4$): Reparar el auto.
  - Acción 5 ($k = 5$): Conducir abollado.

7.2.3. Matriz de costos:

Estado	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
0	0	4.5	5	—	—
1	—	—	—	50	9

7.2.4. Probabilidades de transición:

Desde el estado 0 (auto sano):

- Acción $k = 1$: $p_{00}(1) = \frac{9}{10}$, $p_{01}(1) = \frac{1}{10}$
- Acción $k = 2$: $p_{00}(2) = \frac{49}{50}$, $p_{01}(2) = \frac{1}{50}$
- Acción $k = 3$: $p_{00}(3) = 1$, $p_{01}(3) = 0$

Desde el estado 1 (auto abollado):

- Acción $k = 4$: $p_{10}(4) = 1$, $p_{11}(4) = 0$ (repara y vuelve a estado 0)
- Acción $k = 5$: $p_{10}(5) = 0$, $p_{11}(5) = 1$ (permanece abollado)

7.2.5. Formulación de programación lineal:

Variables de decisión: $y_{01}, y_{02}, y_{03}, y_{14}, y_{15}$

7.2.6. Función objetivo:

$$\text{Minimizar} \quad Z = 4{,}5\, y_{02} + 5\, y_{03} + 50\, y_{14} + 9\, y_{15}$$

7.2.7. Restricción de normalización:

$$y_{01} + y_{02} + y_{03} + y_{14} + y_{15} = 1$$

7.2.8. Restricciones de balance de flujo:

Estado 0 (auto sano):

$$y_{01} + y_{02} + y_{03} = \frac{9}{10}\, y_{01} + \frac{49}{50}\, y_{02} + y_{03} + y_{14}$$

Interpretación: El flujo que sale del estado 0 (estacionando) debe ser igual al flujo que entra (auto que sobrevive sin abolladuras o es reparado).

Estado 1 (auto abollado):

$$y_{14} + y_{15} = \frac{1}{10}\, y_{01} + \frac{1}{50}\, y_{02} + y_{15}$$

Interpretación: El flujo que sale del estado 1 (reparar o conducir abollado) debe ser igual al flujo que entra (auto que se abolla o ya estaba abollado).

7.2.9. No negatividad:

$$y_{ik} \geq 0 \quad \forall\, i, k$$

7.2.10. Interpretación de la conservación de flujo:

Para el estado 0 (auto sano):

$$\underbrace{y_{01} + y_{02} + y_{03}}_{\text{frecuencia de estar sano}} = \underbrace{\frac{9}{10}\, y_{01} + \frac{49}{50}\, y_{02} + y_{03}}_{\text{sobrevive sin abolladuras}} + \underbrace{y_{14}}_{\text{auto reparado}}$$

- **Izquierda**: frecuencia con que el auto está sano y se estaciona.
- **Derecha**: probabilidad de terminar sano al día siguiente (no se abolla o se repara).