Usando la siguiente referencia http://www.statslab.cam.ac.uk/~rrw1/markov/M.pdf y/o cualquiera otra fuente que considere conveniente, responde las siguientes preguntas. Puede subir sus respuestas en un único PDF al Foro

**(2 ptos) Ideas & Conceptos**

1. **¿Qué es un proceso estocástico, que tipos hay y que aplicaciones tienen?**

Según la RAE la palabra estocástico significa del azar, o que se produce por efecto del azar [1] y estas es una de las particularidades de un proceso estocástico que tiene una componente de azar y aleatoriedad.

Más contextualizado a las matemáticas, un proceso estocástico es un concepto matemático que sirve para representar magnitudes aleatorias que varían en función de otra magnitud, generalmente el tiempo. Cada variable aleatoria tiene su función de distribución de probabilidad y pueden o no estar correlacionadas entre sí [2].

Los posibles valores que puede tomar la variable aleatoria se le denominan estados, y pueden ser discretos o continuos, por otro lado, la variable de tiempo puede ser de tipo discreto (cada día, mes, año) de tipo continuo (cualquier instante de tiempo).

En general los procesos estocásticos pueden ser clasificados tal y como se aprecia en la siguiente tabla [3]:

| | t discreto | T continuo |
|---|---|---|
| X discreta | Proceso de estado y tiempo discretos (Cadena) (Unidades producidas mensualmente de un producto) | Proceso de estado discreto y tiempo continuo (Proc. Saltos Puros) (Unidades producidas hasta el instante t) |
| X continua | Proceso de estado continuo y tiempo discreto (Toneladas de producción diaria de un producto) | Proceso de estado y tiempo continuos (Proceso Continuo) (Velocidad de un vehículo en el instante t) |

Tal que X representa variables aleatorias y t el tiempo.

La aplicación fundamental de los procesos estocástico y su modelización es intentar predecir como se comportará en un futuro un sistema dinámico y cambiante con el tiempo que tiene como propiedad intrínseca cierto grado de azar (clima, mercados, redes de comunicaciones, etc.). Así como entenderlos, inferir parámetros ocultos, controlar u optimizar el sistema o cuantificar riesgo. Algunos ejemplos de aplicaciones son:

- **Finanzas:** Movimiento browniano, donde se modela precio de acciones, volatilidad, tasas de interés, etc.
- **Telecomunicaciones y redes:** Cadenas de Markov, donde se modela llegada de paquetes, congestión, fallos de enlace, etc.
- **Epidemiología:** Procesos de ramnificación, SIR estocástico, donde se modela entre otras muchas cosas, propagación de enfermedades, etc.

Esto son solos unos pocos ejemplos de aplicaciones del modelado de los procesos estocásticos.

2. **¿De qué tipo de proceso estocástico son las cadenas de Markov? Describa brevemente estos procesos.**

Las cadenas de Markov son procesos estocásticos cuyas **variables aleatorias son discretas** y el **tiempo se mide de manera discreta**.

La característica más importante es la **propiedad de Markov** (o de *memoria corta*): para predecir el próximo paso basta con conocer el estado actual; la trayectoria completa no aporta información adicional. Formalmente [0]

$$P(X_{n+1} = i_{n+1} | X_0 = i_0, \dots, X_n = i_n) = P(X_{n+1} | X_n = i_n) = p_{i_n i_{n+1}}$$

Esto se lee de manera más informal como la probabilidad de saltar al estado "n +1" conociendo todos los estados anteriores "$X_0 = i_0, \dots, X_n = i_n$" sea igual a la probabilidad de salta al estado "n+1" conociendo únicamente el estado anterior. Es decir, que en realidad solo importe el estado "n" y no todos los estados anteriores.

Una cadena de Markov queda completamente determinada por una distribución inicial $\lambda$ que indica en que estado se empieza y una matriz de transición $P = (p_{ij})$ cuyas filas dan las probabilidades de salto. A partir de estos datos, la distribución de estados evoluciona paso a paso según $v_{n+1} = v_n P^n$

3. **¿Qué aplicaciones tienen en IA los Procesos de Decisión de Markov (MDP)?**

Un Proceso de Decisión de Markov (PDM) (o Markov Decision Process, MDP) extiende la cadena de Markov incorporando acciones —que el agente puede elegir— y recompensas —que cuantifican el beneficio de cada transición—. Es el marco matemático estándar para tomar decisiones secuenciales en entornos estocásticos: el siguiente estado y la recompensa dependen solo del estado actual y de la acción elegida, no del historial completo [4].

En inteligencia artificial, las aplicaciones más destacadas son:

- Aprendizaje por refuerzo [4]: aprender por ensayo-error a tomar buenas decisiones cuando el modelo del mundo es desconocido.
- Robotica y control.
- Sistema de recomendación.
- Procesamiento de dialogo.

**4. ¿Qué es una matriz estocástica (o matriz de transición) y que aplicación tienen en los procesos de Markov?**

Una matriz estocástica es una matriz cuadrada usada para describir las transiciones de cadena de markov [5], esta matriz $P = (p_i j)$ cumple que:

- $p_{\mathrm{ij}} \geq 0 \; \forall i, j$
- La suma de las probabilidades de cada fila es igual a 1.

Sirve en la cadena de Markov para lo siguiente:

- Regla del movimiento: $p_{\mathrm{ij}}$ es la probabilidad de pasar de lestado i al estado j en un solo paso.
- Actualización de la distribución: Si $v_n$es el vector fila de probabilidades en el instante “n”, entonces $v_{n+1} = v_n \cdot P$
- Transiciones de varios pasos: El elemento (i,j) de $P^k$ da la probabilidad de ir de i a j en exactamente “k” pasos [3]

En resumen, la matriz estocástica **contiene toda la dinámica** de la cadena de Markov: especifica cómo se mueve el sistema, permite calcular su evolución y revela su comportamiento estable.

**5. ¿Qué es el vector de estado de un proceso de Markov, y qué representa?**

Para un instante “n”, el vector de estado (también llamada vector de distribución) es:

$$v_n = (Pr[X_n = s_1], Pr[X_n = s_2], \dots, Pr[X_n = S_m])$$

Donde $s_1, s_2, \dots, s_n$ son los estados posibles. Cada componente de $v_n^{(i)}$ indica la probabilidad de que el proceso este en el estado $s_i$ en ese paso. Además, debe cumplir que:

- Todos sus elementos sean mayores o iguales que cero.
- La suma de todos los elementos sea igual a 1.

6. **¿Qué es el estado estacionario de una cadena de Markov con un número finito de estados? ¿Cómo se calcula? Y ¿Qué significado tiene?**

   El estado estacionario de una cadena de Markov con un número finito de estados es una distribución de probabilidad que permanece constante a lo largo del tiempo. Es decir, si el sistema alcanza esta distribución, se mantendrá en ella en todos los pasos futuros, alcanzando un estado de equilibrio

   Se calcula resolviendo la ecuación $\pi \cdot P = \pi$ [6], es decir, obteniendo el vector que multiplicado por la matriz de transición de ese mismo vector.

   El estado estacionario es el **equilibrio probabilístico** al que la cadena de Markov termina llegando —y en el que permanecería para siempre— independientemente de dónde empezó. Es decir, con el paso del tiempo la influencia del estado inicial se diluye.

**Modelo del clima – Generalidades**

Considera un modelo del clima con tres estados posibles: (1) Soleado, (2) Nublado y (3) Lluvioso.

1. **¿Pueden las filas de la siguiente matriz, se las filas de una matriz de transición de un tal proceso de Markov?**

$$Q = \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.1 & 0.5 & 0.3 \\ 0.7 & 0.5 & \text{-}0.2 \end{pmatrix}$$

Tal y como se comento en la pregunta 4 del anterior apartado debe cumplirse dos cosas para cada fila:

- Que todos los elementos sean mayor o igual a cero. En la primera y segunda fila se cumple, pero en la tercera fila no, así que la podemos descartar.
- Que la suma de probabilidades de cada fila sea igual a 1. No se cumple en ninguna fila.

Teniendo en cuenta el análisis de las dos anteriores premisas la respuesta es No, no pueden ser fila de una matriz de un tal proceso de Markov.

2. **¿Puede alguno de los vectores siguientes, se ele vector de estado actúa de un tal proceso de Markov?**

$$u_0 = (0.3{,}0.6{,}0.7) \quad w_0 = (0.2{,}0.1{,}0.5)$$

Nuevamente debe cumplirse que todos los valores sean mayores o iguales que cero y que su suma de uno, en ninguno de los dos vectores se cumple que la suma sea igual a 1, por lo que no pueden ser vectores de estado de un tal proceso de markov.

## (5 ptos) Modelo del Clima – Ejercicio

Considere un modelo del clima con tres estados posibles: (1) Soleado, (2) Nublado y (3) Lluvioso, con matriz de transición P:

$$\mathrm{P} = \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix}$$

Si el vector de estado actual es $v_0 = \left(P(S_0), P(N_0), P(L_0)\right) = (0.75, 0.2, 0.05)$ calcule:

1. **El vector de estado para mañana $\boldsymbol{v = (P(S_0), P(N_0), P(L_0))}$**

$$v_{\mathrm{n+1}} = v_n \cdot P$$

$$v_{\mathrm{n+1}} = (0.75, 0.2, 0.05) \cdot \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix}$$

$$v_{\mathrm{n+1}} = \begin{pmatrix} 0{,}75 \cdot 0{,}2 + 0{,}2 \cdot 0{,}1 + 0{,}05 \cdot 0{,}5 \\ 0{,}75 \cdot 0{,}5 + 0{,}2 \cdot 0{,}6 + 0{,}05 \cdot 0{,}3 \\ 0{,}75 \cdot 0{,}3 + 0{,}2 \cdot 0{,}3 + 0{,}05 \cdot 0{,}2 \end{pmatrix} = \begin{pmatrix} 0{,}195 \\ 0{,}51 \\ 0{,}295 \end{pmatrix}$$

**Nota:** $v_{\mathrm{n+1}}$ es un vector fila, lo que, por motivos de espacio, se ha decidido representarlo como columna.

La probabilidad de que este soleado es del 19,5%, de que este Nublado un 51% y de que este lluvioso un 29,5%, como podemos apreciar, el nuevo vector de estado sigue manteniendo la propiedad de que la suma sea igual a 1.

2. **Los vectores de estado para pasado mañana y de hoy a tres días.**

$$v_{n+2} = v_{n+1} \cdot P = (0{,}195 \quad 0{,}51 \quad 0{,}295) \cdot \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix} = \begin{pmatrix} 0{,}2375 \\ 0{,}4920 \\ 0{,}2705 \end{pmatrix}$$

$$v_{n+3} = v_{n+2} \cdot P = (0{,}2375 \quad 0{,}4920 \quad 0{,}2705) \cdot \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix} = \begin{pmatrix} 0{,}23195 \\ 0{,}49510 \\ 0{,}27295 \end{pmatrix}$$

**Nota:** nuevamente los vectores finales se han representado como vectores columna por motivos de espacio, pero en realidad, son vectores filas.

### 3. Calcule el estado estacionario de la matriz de transición P.

$$\pi \cdot P = \pi$$

$$(\mathrm{x} \quad \mathrm{y} \quad \mathrm{z}) \cdot \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix} = (\mathrm{x} \quad \mathrm{y} \quad \mathrm{z})$$

$$\begin{cases} 0.2x + 0.1y + 0.5z = \mathrm{x} \\ 0.5x + 0.6y + 0.3z = \mathrm{y} \\ 0.3\mathrm{x} + 0.3\mathrm{y} + 0.2\mathrm{z} = \mathrm{z} \\ \mathrm{x} + \mathrm{y} + \mathrm{z} = 1 \end{cases}$$

Además de las dos ecuaciones se ha añadido la condición de normalización, es decir, que la suma de las probabilidades sea 1. Si seleccionamos las dos primeras ecuaciones y la última, tendremos la siguiente matriz que representa nuestro sistema de ecuaciones:

$$\left(\begin{array}{ccc|c} -0.8 & 0.1 & 0.5 & 0 \\ 0.5 & -0.4 & 0.3 & 0 \\ 1 & 1 & 1 & 1 \end{array}\right)$$

Si aplicamos en Python la solución a este sistema tendremos que

```
A = np.array([[0.8, -0.1, -0.5],
              [-0.5, 0.4, -0.3],
              [1.0, 1.0, 1.0]], dtype=float)
b = np.array([0.0, 0.0, 1.0])

# Resolvemos A·π = b
pi = np.linalg.solve(A, b)

print("Estado estacionario π =", pi)
print("Comprobación: suma =", pi.sum())
```

[2] ✓ 0.0s

```
Estado estacionario π = [0.23232323 0.49494949 0.27272727]
Comprobación: suma = 1.0
```

### 4. ¿Cuál debería ser el vector de estado actual para que mañana el vector de estado sea $v = (0.75, 0.2, 0.05)$?

$$(0{,}75 \quad 0{,}2 \quad 0{,}05) = (\mathrm{x} \quad \mathrm{y} \quad \mathrm{z}) \cdot \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix}$$

$$\begin{cases} 0.2x + 0.1y + 0.5z = 0{,}75 \\ 0.5x + 0.6y + 0.3z = 0{,}2 \\ 0.3\mathrm{x} + 0.3\mathrm{y} + 0.2\mathrm{z} = 0{,}05 \end{cases}$$

Nuevamente aplicamos este sistema en Python dándonos como resultado:

```
A = np.array([[0.2, 0.1, 0.5],
              [0.5, 0.6, 0.3],
              [0.3, 0.3, .20]], dtype=float)
b = np.array([0.75, 0.2, 0.05])

sol = np.linalg.solve(A, b)

print("Solución", sol)
```

```
[4]  ✓ 0.0s
...  Solución [-3.5  2.   2.5]
```

Como podemos apreciar nos dan números negativos y la suma de las probabilidades no da 1, por lo que no existe un vector de estado actual que nos lleve al estado de mañana propuesto en el enunciado.

5. Si la entrada $ij$ de la matriz P la denotamos por $p_{ij}$, donde los índices i,j corresponde a los estados i y j respectivamente (1 para soleado, 2 para nublado y 3 para lluvioso)
   a. **¿Qué representa la probabilidad $p_{ij}$ en términos de los estados i y j?**

   La probabilidad $p_{\mathrm{ij}}$ representa la probabilidad de que se transite del estado "i" al estado "j". Por ejemplo, para la siguiente matriz

$$\begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.6 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{pmatrix}$$

   La probabilidad de que estando lluvioso se pase al estado soleado es del 50%, este dato lo podemos ver en la fila 3 columna 1.

b. Si $S_0, N_0 y L_0$ corresponden a los estados actuales, y S, N y L a los estados de mañana, identifique con las entradas de la matriz P las siguientes probabilidades:

   i. $\boldsymbol{P(S|S_0)}$**:** La probabilidad de que mañana este soleado dado que hoy lo está.

   En este caso sería el valor especificado en la fila 1 columna 1, que es del 0.2, es decir, hay un 20% de probabilidad de que estando hoy soleado mañana también lo esté.

   ii. $\boldsymbol{P(N|L_0)}$**:** La probabilidad de que mañana este nublado dado que hoy esta lloviendo. Sería la fila 3 columna 2, es decir, que tiene un valor del 0.3

   iii. $\boldsymbol{P(L|N_0)}$: La probabilidad de que mañana llueva dado que hoy este nublado sería la fila 2 columna 3, nuevamente una probabilidad del 0.3

**(2 ptos) Tensores – Resumen**

1. **Resuma las ideas principales del artículo que encontrará en el siguiente link: https://arxiv.org/pdf/2010.03313**

El problema subyacente es que librerías como TensorFlow, PyTorch y JAX tienen problemas de eficiencia calculando las derivadas de matrices en forma de Jacobianos o Hessianos. El núcleo de esta deficiencia es el lenguaje matemático subyacente que se usa para calcular estas derivadas.

Existe un algoritmo muy rápido para este cálculo, pero se basa en la **notación de Ricci**, que es incompatible con los *frameworks* populares de *deep learning*, los cuales usan la **notación de Einstein**, que es más simple. Cambiar la notación en todos estos *frameworks* es una opción impracticable.

Los autores desarrollan un nuevo método para el cálculo de derivadas tensoriales que utiliza la **notación de Einstein**, eliminando el obstáculo de la incompatibilidad. En su versión básica, este nuevo método es tan eficiente como el enfoque anterior basado en la notación de Ricci.

Las mejoras clave son:

- **Modo "Cross-Country"**: se reordenan las multiplicaciones para hacerlas más eficientes. En lugar de multiplicar en un orden fijo, la estrategia multiplica primero los tensores de menor orden (vectores con vectores, luego matrices, etc.), lo que reduce significativamente el tiempo de cálculo.
- **Compresión de derivadas**: La representación de derivadas de orden superior se hace de forma mucho más compacta. Por ejemplo, un hessiano que normalmente sería un tensor de cuarto orden puede ser comprimido a un tensor de tercer orden o incluso a una matriz, lo que acelera drásticamente los cálculos.

Tras aplicar estas mejoras los resultados son:

- Rendimiento superior siendo varios ordenes de magnitud más rápido.
- Gracias a la nueva eficiencia a viabilidad de nuevos cálculos.

Además, los autores han implementado estos algoritmos en la herramienta en línea **www.MatrixCalculus.org**, poniendo la tecnología a disposición de la comunidad.

**Referencias**

**[0]** http://www.statslab.cam.ac.uk/~rrw1/markov/M.pdf

[1] https://www.rae.es/diccionario-estudiante/estoc%C3%A1stico

[2] https://es.wikipedia.org/wiki/Proceso_estoc%C3%A1stico

[3]https://www.dmae.upct.es/~mcruiz/Telem06/Teoria/apuntes_procesos.pdf#:~:text=Una%20Cadena%20es%20un%20proceso%20estoc%C3%A1stico%20en,valores%20discretos%20en%20el%20espacio%20de%20estados.

[4] https://es.wikipedia.org/wiki/Proceso_de_decisi%C3%B3n_de_M%C3%A1rkov

[5] https://en.wikipedia.org/wiki/Stochastic_matrix

[6] https://brilliant.org/wiki/stationary-distributions/