Modelos de Markov

...

Algoritmos y Estructuras de Datos
2024

UNRaf UNIVERSIDAD NACIONAL DE RAFAELA

Sistemas determinísticos vs estocásticos

- Se denomina sistema **determinístico** a aquel en que el azar no está involucrado en el desarrollo de los futuros estados del sistema.
- En un modelo determinístico, un mismo conjunto de condiciones de partida producirán **siempre** la misma salida.
- Por su parte, en los sistemas **estocásticos** la salida está afectada por **elementos aleatorios**.
- En los sistemas estocásticos, cada salida está vinculada a una **probabilidad** de ocurrencia.

Modelos de Markov

- En los procesos estocásticos es posible **predecir** cuál sería el **estado futuro más probable** de un sistema en base a todos los estados anteriores.

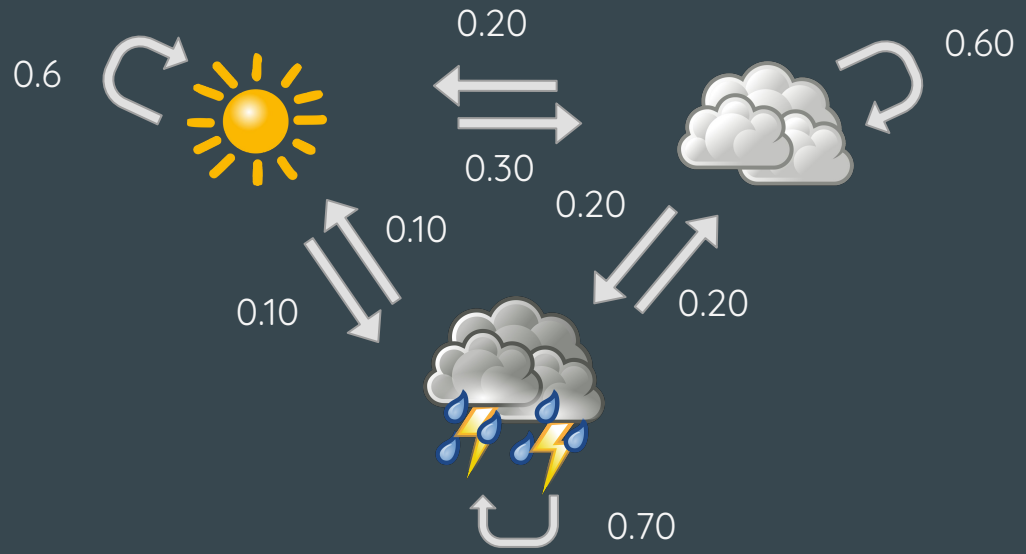
$$P(X_{t+1} = y | X_0 = x_0, X_1 = x_1, \ldots, X_t = x_t)$$

- Para que un proceso estocástico se considere "**markoviano**", la probabilidad condicional de un **estado futuro** depende **únicamente** del **estado actual.**

$$P(X_{t+1} = y | X_0 = x_0, X_1 = x_1, \ldots, X_t = x_t) = P(X_{t+1} = y | X_t = x_t)$$

- Este supuesto permite reducir significativamente el número de cálculos necesarios para cada predicción.

Modelos de Markov



- Un ejemplo típico de modelos markovianos es la predicción del tiempo.
- Al plantearlo como un modelo de Markov, no importa que haya pasado en días anteriores.

Modelos de Markov: definiciones

- **Estado**: cada una de las posibles valores que puede tomar el sistema
- **Espacio de estados**: todos los estados presentes en el sistema
- **Probabilidad de transición**: la probabilidad de pasar de un estado a otro.

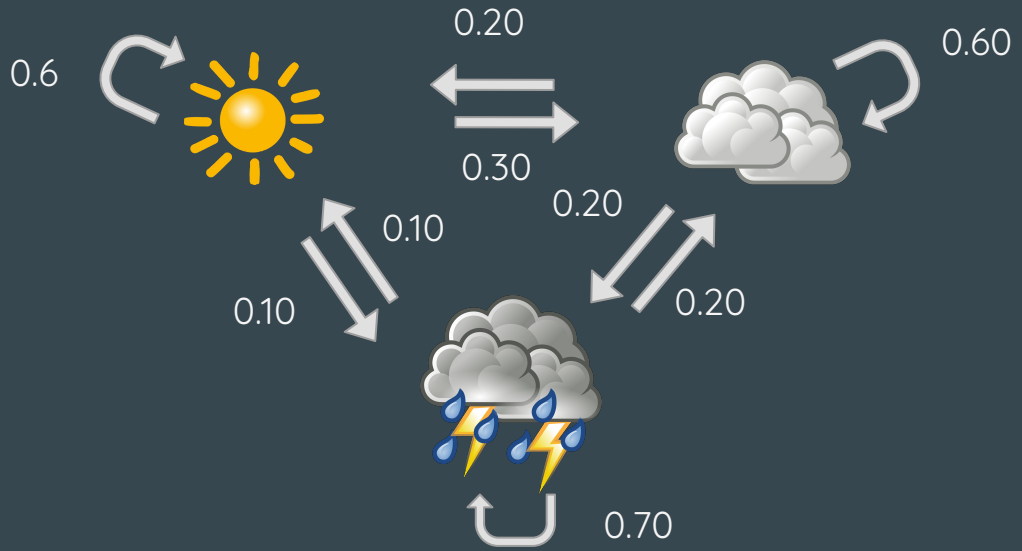
Tipos de modelos de Markov

- Los modelos se pueden clasificar en base a si los estados son **totalmente observables** o si solo pueden verse las **emisiones** de un estado.
- Por ejemplo, predecir el nivel de tráfico (no observable) sólo a partir del clima (observable).
- También pueden clasificarse dependiendo de si las probabilidades para pasar de un estado a otro dependen de un agente externo (controlado) o no (autónomo)

	Estados totalmente observables	**Estados parcialmente observables**
Sistema autónomo	Cadenas de Markov	Modelos ocultos de Hidden Markov
Sistema controlado	Procesos de decisión de Markov	Procesos de decisión de Markov parcialmente observables

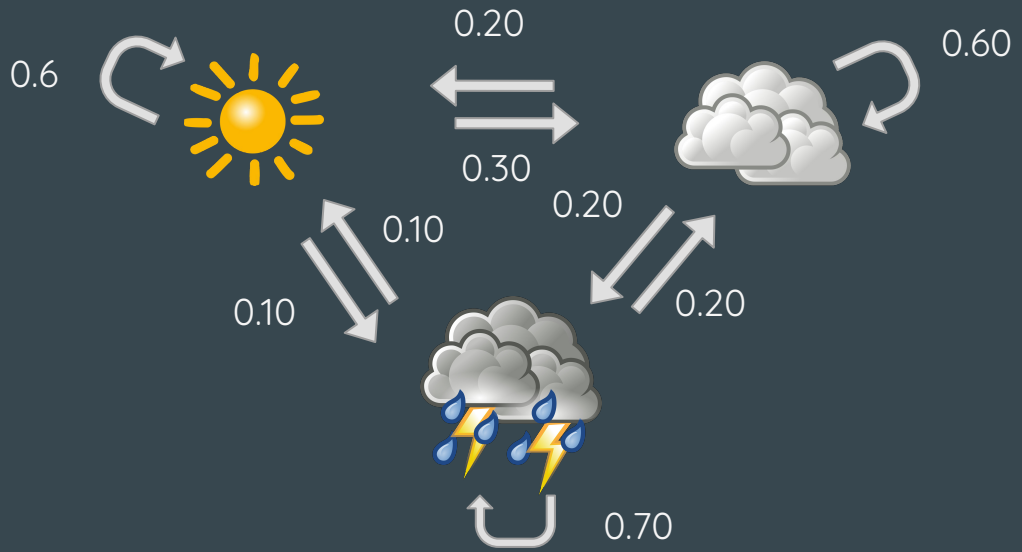
Cadenas de Markov

- Son los modelos de Markov más simples, ya que todos los estados son observables
- Los cambios entre estados se denominan **transiciones**; cada **transición** tiene asociada una **probabilidad de transición**.



Cadenas de Markov

- Estos datos pueden expresarse como una serie de **probabilidades condicionales**.
- La suma de todas las probabilidades a partir de un estado debe ser 1.



- P(sol/sol) = 0.60
- P(lluvia/sol) = 0.10
- P(nube/sol) = 0.30

} 1

- P(sol/lluvia) = 0.10
- P(lluvia/lluvia) = 0.70
- P(nube/lluvia) = 0.20

} 1

- P(sol/nube) = 0.20
- P(lluvia/nube) = 0.20
- P(nube/nube) = 0.60

} 1

Cadenas de Markov

- También pueden plantearse como una **matriz de transición**.
- Cada fila representa un estado (X_t) y las probabilidades de pasar a otros estados en el futuro (X_{t+1}).
- La matriz de transición siempre tendra tamaño N x N, siendo N el número de estados del sistema

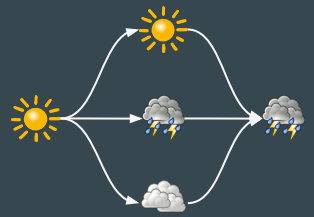
$$P = \begin{array}{c} \\ \\ \\ \end{array} \overset{X_{t+1}}{\begin{pmatrix} 0.60 & 0.10 & 0.30 \\ 0.10 & 0.70 & 0.20 \\ 0.20 & 0.20 & 0.60 \end{pmatrix}} \begin{array}{l} \\ \\ \\ \end{array} X_t$$

Cadenas de Markov: predicción

- A partir de un estado inicial, es posible calcular las probabilidades de cualquier estado futuro.
- Por ejemplo, si el estado inicial es soleado, la probabilidad de que llueva al día siguiente se puede obtener de la matriz.

$$P(\text{lluvia/sol}) = 0.10$$

- Si quisiéramos saber la probabilidad de que llueva dentro de 2 días, debemos considerar todos los estados intermedios



$$P(\text{sol/sol})*P(\text{lluvia/sol}) + P(\text{lluvia/sol})*P(\text{lluvia/lluvia}) + P(\text{nube/sol})*P(\text{lluvia/nube}) = 0.6*0.1 + 0.1*0.7 + 0.3*0.2 = 0.19$$

Cadenas de Markov: predicción

- Otra forma de expresar los cálculos es multiplicando la matriz de transición por un vector con las probabilidades de un estado.
- Suponiendo que el estado actual es “sol”, podríamos expresar las probabilidades como:

$$X_t = [1, 0, 0]$$

- Para calcular X_{t+1} debemos multiplicar este vector por la matriz de transición:

$$[1, 0, 0] \times \begin{pmatrix} 0.60 & 0.10 & 0.30 \\ 0.10 & 0.70 & 0.20 \\ 0.20 & 0.20 & 0.60 \end{pmatrix} = [1*0.6, 1*0.1, 1*0.3] = [0.6, 0.1, 0.3]$$

Cadenas de Markov: predicción

- Si quisiéramos calcular el estado siguiente (X_{t+2}), debemos multiplicar el nuevo vector X_{t+1} por la matriz de transición:

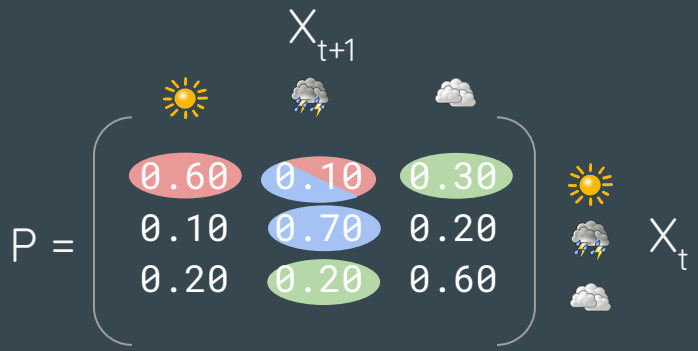
$$X_{t+2} = X_{t+1} \times P$$

$$X_{t+2} = [0.6,\ 0.1,\ 0.3] \times \begin{pmatrix} 0.60 & 0.10 & 0.30 \\ 0.10 & 0.70 & 0.20 \\ 0.20 & 0.20 & 0.60 \end{pmatrix} = [0.43,\ 0.19,\ 0.38]$$

- Generalizando, la probabilidad de un estado X_{n+1} se puede calcular como la el vector X_n por la matriz de transición

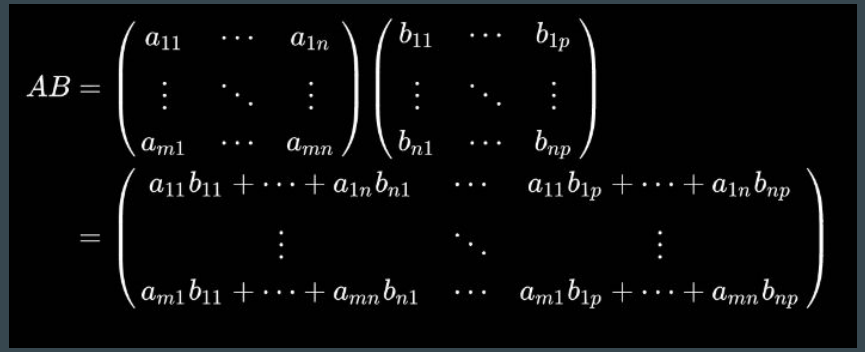
Cadenas de Markov: generalización

- Para definir la probabilidad en el día 2 de un estado a otro, se multiplica los valores de la “fila objetivo” con los valores de la “columna de inicio”.



- Para calcular la probabilidad de cualquier estado futuro a partir de un estado presente, debemos multiplicar la fila y la columna pertinentes.
- Para tener todas las probabilidades de X_2, es necesario multiplicar la matriz por sí misma. Para cualquier otro dia, se debe hacer X_n x P.

Multiplicación de matrices: pseudocódigo



Entrada: **matriz A (tamaño m*n) y matriz B (tamaño n*p)**

```
Recorrer una fila de una matriz (m):
    Recorrer una columna de la otra matriz (p):
        Recorrer todas las columnas de la primer matriz y las filas
        de la segunda (n):
            Resultado[m][p] += A[m][n] * B[n][p]
```

Multiplicación de matrices: código

```
def multiplicacion_matrices(A, B):
    resultado = []
    filas_A = len(A)
    columnas_A = len(A[0])
    columnas_B = len(B[0])
    for i in range(filas_A):
        resultado.append([])
        for j in range(columnas_B):
            celda = 0
            for k in range(columnas_A):
                celda += A[i][k] * B[k][j]
            resultado[i].append(celda)
    return resultado

A = [[2,3], [4,5], [2,1]]
B = [[4,3,2,1], [5,6,7,8]]
C = multiplicacion_matrices(A, B)
```

Distribución estacionaria

- Una de las cosas interesantes para buscar en las cadenas de Markov es la existencia de **distribuciones estacionarias** a partir de un estado inicial.
- Las distribuciones estacionarias son conjuntos de probabilidades (π) que ya no cambian en el tiempo, por más que se los multiplique por la matriz de transición.

$$\pi \times P = \pi$$

- Para calcular esta matriz existen 2 estrategias:
  - Cálculo de autovectores (exacto)
  - Cálculo por fuerza bruta de iteraciones hasta lograr un conjunto de probabilidades igual al anterior (aproximado y en caso de no existir, itera infinitamente).

Distribución estacionaria: código

```
x0 = [[1, 0 ,0]] # Estado inicial: sol
P = [[0.6, 0.1, 0.3], [0.1, 0.7, 0.2], [0.2, 0.2, 0.6]]
x1 = multiplicacion_matrices(x0, P)
contador = 1
while x0 != x1:
    x0 = x1
    x1 = multiplicacion_matrices(x0, P)
    contador += 1

print(x1)
print("Numero de iteraciones:", contador)
```