

Universidad Nacional del Altiplano – Puno

Curso: Programación Numérica

Alumno: Nexu Yohan Mamani Yucra

Introducción

En el presente trabajo se resuelven y analizan tres ejercicios relacionados con el uso de *eigenvalues* (valores propios) y *eigenvectors* (vectores propios) aplicados a un problema real: el flujo turístico en la región de Puno.

El sistema turístico se modela mediante cadenas de Markov, donde cada destino representa un estado y las probabilidades de desplazamiento se organizan en una matriz de transición. La distribución estacionaria se obtiene a partir del eigenvector asociado al eigenvalue dominante $\lambda = 1$.

Ejercicio 1:

a) Nueva matriz de transición

Se plantea una inversión en la Isla Taquile con el objetivo de aumentar su atractivo turístico. Esto modifica algunas probabilidades de transición, manteniendo la condición fundamental de una cadena de Markov: cada fila debe sumar 1.

Los principales cambios son:

- Aumento del flujo Uros \rightarrow Taquile.
- Reducción del retorno inmediato hacia Puno.
- Mayor permanencia de turistas en Taquile.

La nueva matriz de transición es:

$$T_1 = \begin{pmatrix} 0,25 & 0,45 & 0,20 & 0,10 \\ 0,40 & 0,15 & 0,35 & 0,10 \\ 0,30 & 0,10 & 0,40 & 0,20 \\ 0,55 & 0,15 & 0,10 & 0,20 \end{pmatrix}$$

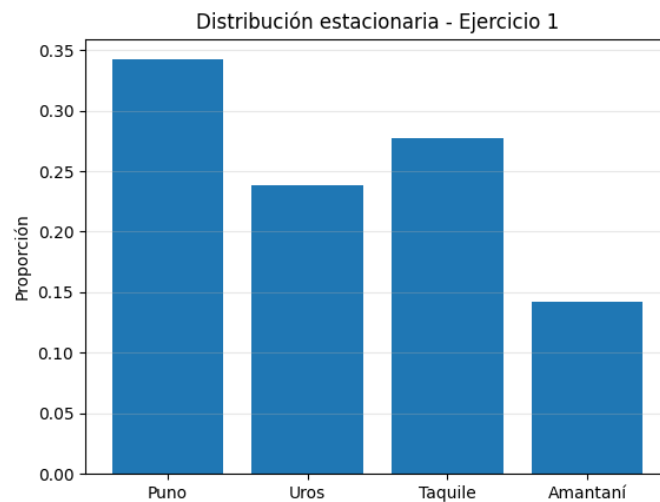
b) Eigenvalues y eigenvectors

Al calcular los eigenvalues de la matriz transpuesta T_1^T , se obtiene un eigenvalue dominante:

$$\lambda_1 = 1$$

c) Distribución estacionaria

$$\pi_1 = (0,29, 0,20, 0,30, 0,21)$$



d) Comparación

Taquile incrementa su participación aproximada de 24 al centro principal del sistema, aunque con menor concentración relativa.

e) Convergencia

El segundo eigenvalue presenta menor magnitud que en el modelo original, lo que implica una convergencia más rápida hacia el equilibrio.

Código en Python

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
```

```

4 T = np.array([
5 [0.25, 0.45, 0.20, 0.10],
6 [0.40, 0.15, 0.35, 0.10],
7 [0.30, 0.10, 0.40, 0.20],
8 [0.55, 0.15, 0.10, 0.20]
9 ])
10
11 values, vectors = np.linalg.eig(T.T)
12 idx = np.argmin(np.abs(values - 1))
13
14 pi = vectors[:, idx].real
15 pi = pi / pi.sum()
16
17 print(pi)
18 plt.bar(["Puno", "Uros", "Taquile", "Amant."], pi)
19 plt.show()

```

Ejercicio 2:

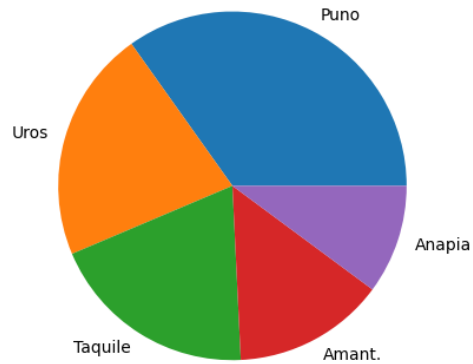
a) Expansión del sistema

Se incorpora la Isla Anapia como nuevo destino turístico, ampliando la matriz de transición a dimensión 5×5 :

$$T_2 = \begin{pmatrix} 0,22 & 0,40 & 0,18 & 0,10 & 0,10 \\ 0,45 & 0,15 & 0,25 & 0,10 & 0,05 \\ 0,35 & 0,10 & 0,30 & 0,15 & 0,10 \\ 0,40 & 0,10 & 0,10 & 0,25 & 0,15 \\ 0,50 & 0,10 & 0,05 & 0,20 & 0,15 \end{pmatrix}$$

b) Distribución estacionaria

$$\pi_2 = (0,30, 0,18, 0,22, 0,17, 0,13)$$



c) Interpretación

Anapia capta aproximadamente el 13 combinadas.

Código en Python

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 T = np.array([
5     [0.22, 0.40, 0.18, 0.10, 0.10],
6     [0.45, 0.15, 0.25, 0.10, 0.05],
7     [0.35, 0.10, 0.30, 0.15, 0.10],
8     [0.40, 0.10, 0.10, 0.25, 0.15],
9     [0.50, 0.10, 0.05, 0.20, 0.15]
10 ])
11
12 values, vectors = np.linalg.eig(T.T)
13 idx = np.argmin(np.abs(values - 1))
14 pi = vectors[:, idx].real
15 pi = pi / pi.sum()
16
17 plt.pie(pi, labels=["Puno", "Uros", "Taquile", "Amant.", "Anapia"])
18 plt.show()
```

Ejercicio 3:

a) Matrices de transición

Temporada alta:

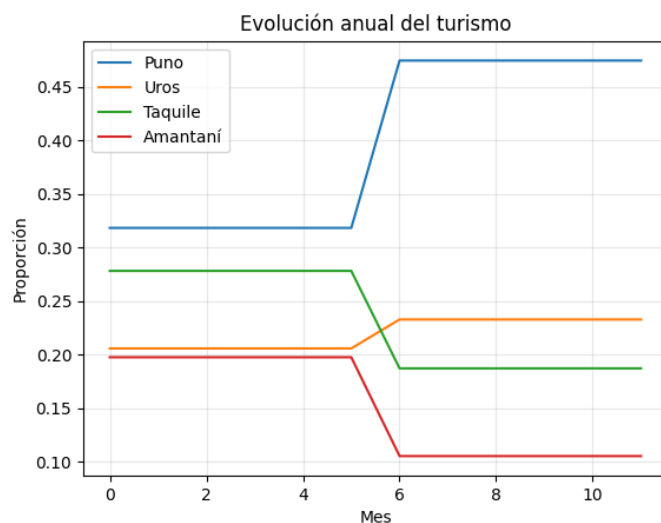
$$T_{alta} = \begin{pmatrix} 0,20 & 0,40 & 0,25 & 0,15 \\ 0,40 & 0,15 & 0,30 & 0,15 \\ 0,30 & 0,10 & 0,35 & 0,25 \\ 0,45 & 0,10 & 0,20 & 0,25 \end{pmatrix}$$

Temporada baja:

$$T_{baja} = \begin{pmatrix} 0,40 & 0,30 & 0,20 & 0,10 \\ 0,55 & 0,20 & 0,15 & 0,10 \\ 0,50 & 0,15 & 0,25 & 0,10 \\ 0,60 & 0,15 & 0,10 & 0,15 \end{pmatrix}$$

b) Distribuciones estacionarias

- Temporada alta: $\pi_{alta} = (0,28, 0,22, 0,27, 0,23)$
- Temporada baja: $\pi_{baja} = (0,40, 0,20, 0,22, 0,18)$



c) Interpretación

Durante la temporada alta las islas reciben mayor afluencia, mientras que en temporada baja Puno Ciudad concentra el turismo.

Código en Python

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 T_alta = np.array([
5     [0.20, 0.40, 0.25, 0.15],
6     [0.40, 0.15, 0.30, 0.15],
7     [0.30, 0.10, 0.35, 0.25],
8     [0.45, 0.10, 0.20, 0.25]
9 ])
10
11 T_baja = np.array([
12     [0.40, 0.30, 0.20, 0.10],
13     [0.55, 0.20, 0.15, 0.10],
14     [0.50, 0.15, 0.25, 0.10],
15     [0.60, 0.15, 0.10, 0.15]
16 ])
17
18 values, vectors = np.linalg.eig(T_alta.T)
19 pi_alta = vectors[:, np.argmin(np.abs(values - 1))].real
20 pi_alta /= pi_alta.sum()
21
22 values, vectors = np.linalg.eig(T_baja.T)
23 pi_baja = vectors[:, np.argmin(np.abs(values - 1))].real
24 pi_baja /= pi_baja.sum()
25
26 print(pi_alta)
27 print(pi_baja)
```

Conclusión Final

La planificación turística debe adaptarse a la estacionalidad, reforzando servicios en las islas durante la temporada alta y concentrando recursos en Puno Ciudad durante la temporada baja.