Proyecto 2

Profesor: Cristóbal Rojas

Ayudantes: Fernando de Diego
Jean-Christophe Labarthe

Introducción

En este proyecto pondrás a prueba lo aprendido sobre sistemas de ecuaciones lineales y matrices, en aplicaciones concretas. En particular, desarrollarás la clase `Mat` entregada en el material de *Coding the Matrix* (`http://resources.codingthematrix.com/`), y utilizarás dicha clase para analizar modelos de dinámica de poblaciones. Ten en consideración que la clase `Mat` hace uso de la clase `Vec` que desarrollaste en el proyecto 1. Todos los archivos requeridos están disponibles en el Drive del curso.

Matrices generalizadas

Al igual que los vectores generalizados presentados durante el curso, es posible definir matrices generalizadas, en donde los dominios de las filas y columnas no son necesariamente numéricos (ni ordenados). Formalmente, definimos una $R \times C$-matriz sobre $\mathbb{F}$ como una función desde el espacio producto $R \times C$ a $\mathbb{F}$. Nos referimos a los elementos de R como las etiquetas de las filas y a los elementos de C como las etiquetas de las columnas. Como ejemplo, consideremos $R = \{$`a, b`$\}$ y $C = \{$`#, @, ?`$\}$, y definamos la matriz M como

	@	#	?
a	1	2	3
b	10	20	30

Formalmente, esta matriz es una función desde $R \times C$ a $\mathbb{R}$. Podemos representarla como un diccionario de `Python` de la forma:

```
{(a, @): 1, (a, #): 2, (a, ?): 3, (b, @): 10, (b, #): 20, (b,?): 30}
```

Si consideramos cada fila de la matriz como un vector, podemos representarla como un diccionario de vectores fila, de la siguiente manera:

```
{a: Vec({#, @, ?}, {@: 1, #: 2, ?: 3}),
b: Vec({#, @, ?}, {@: 10, #: 20, ?: 30})}
```

Análogamente, podemos representarla como un diccionario de vectores columna como:

```
{#: Vec({a, b}, {a: 2, b: 20}),
@: Vec({a, b}, {a: 1, b: 10}),
?: Vec({a, b}, {a: 3, b: 30})}
```

En el archivo `matutil.py` se pueden encontrar las funciones `mat2rowdict(M)` y `mat2coldict(M)`, que transforman matrices en sus representaciones como diccionarios de vectores fila y columna, respectivamente.

Dos operaciones especialmente importantes respecto a las matrices generalizadas son el producto matriz-vector y el producto vector-matriz.

Producto matriz-vector

Corresponde a la generalización de la tradicional "multiplicación por la derecha" entre una matriz y un vector, la cual simboliza una combinación lineal de las columnas. Por ejemplo,

$$\begin{aligned} \begin{pmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{pmatrix} * \begin{pmatrix} 7 & 0 & 4 \end{pmatrix} &= 7 \begin{pmatrix} 1 & 10 \end{pmatrix} + 0 \begin{pmatrix} 2 & 20 \end{pmatrix} + 4 \begin{pmatrix} 3 & 30 \end{pmatrix} \\ &= \begin{pmatrix} 19 & 190 \end{pmatrix} \end{aligned}$$

es una multiplicación matriz-vector tradicional. El caso generalizado se puede ver de la forma

$$\begin{array}{c|ccc} & @ & \# & ? \\ \hline a & 2 & 1 & 3 \\ b & 20 & 10 & 30 \end{array} * \begin{array}{ccc} @ & \# & ? \\ \hline 0.5 & 5 & \text{-1} \end{array} = 0.5 \begin{array}{cc} a & b \\ \hline 2 & 20 \end{array} + 5 \begin{array}{cc} a & b \\ \hline 1 & 10 \end{array} \text{ - } \begin{array}{cc} a & b \\ \hline 3 & 30 \end{array} = \begin{array}{cc} a & b \\ \hline 3 & 30 \end{array}$$

Producto vector-matriz

Análogamente al caso anterior, el producto vector-matriz corresponde a la generalización de la tradicional "multiplicación por la izquierda" entre un vector y una matriz, la cual simboliza una combinación lineal de las filas. Por ejemplo,

$$\begin{aligned} \begin{pmatrix} 3 & 4 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{pmatrix} &= 3 \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} + 4 \begin{pmatrix} 10 & 20 & 30 \end{pmatrix} \\ &= \begin{pmatrix} 43 & 86 & 129 \end{pmatrix} \end{aligned}$$

es una multiplicación vector-matriz tradicional. El caso generalizado se puede ver de la forma

$$\begin{array}{cc} a & b \\ \hline 2 & 3 \end{array} * \begin{array}{c|ccc} & @ & \# & ? \\ \hline a & 2 & 1 & 3 \\ b & 20 & 10 & 30 \end{array} = 2 \begin{array}{ccc} @ & \# & ? \\ \hline 2 & 1 & 3 \end{array} + 3 \begin{array}{ccc} @ & \# & ? \\ \hline 10 & 20 & 30 \end{array} = \begin{array}{ccc} @ & \# & ? \\ \hline 34 & 62 & 96 \end{array}$$

Para más información de la teoría detrás de las matrices tradicionales y generalizadas se recomienda fuertemente revisar el capítulo 4 de *Coding the Matrix*.

Clase `Mat` (20 %)

Tal como en el proyecto 1 se implementó la clase `Vec`, en este proyecto se desarrollará la clase `Mat`, que representa matrices generalizadas.

El archivo `mat.py` (disponible en el Drive) contiene el esqueleto de una clase `Mat` que, como su nombre lo indica, permite crear y manipular matrices generalizadas. Para usar esta clase primero debes importarla (`from mat import Mat`). Para instanciar un objeto `Mat`, debes entregar como primer argumento una 2-tupla con el dominio de las filas y el de las columnas. Luego, como segundo argumento debes entregar un diccionario que mapea las tuplas `(fila, columna)` a cada elemento. Por ejemplo, la matriz

$$\begin{array}{c|ccc} & @ & \# & ? \\ \hline a & 1 & 2 & 3 \\ b & 10 & 20 & 30 \end{array}$$

se define de la forma

$M =$ `Mat(({a, b}, {@, #, ?}),`
`{(a, @): 1, (a, #): 2, (a, ?): 3, (b, @): 10, (b, #): 20, (b, ?): 30})`

Recuerde que, para ahorrar memoria, usamos la convención de no incluir en el diccionario los pares *llave:valor* cuando el valor es 0.

Tu primera tarea consiste en rellenar los métodos de la clase `Mat` especificados en el archivo `mat.py`. Estos métodos son:

- `getitem(M, k)`
- `equal(A, B)`
- `setitem(M, k, val)`
- `scalar_mul(M, x)`
- `transpose(M)`
- `vector_matrix_mul(v, M)`
- `matrix_vector_mul(M, v)`
- `matrix_matrix_mul(A, B)`

Una descripción detallada de cada uno, junto al retorno exigido, se muestran en los `docstrings` de los métodos (las descripciones que se entregan inmediatamente después de definir cada función). A continuación, se presentan algunas consideraciones que es necesario tener en mente:

- Cada método incluye una línea de tipo `assert`. Dicha expresión se encarga de asegurar que la condición que le sigue se cumpla. Por ejemplo, la expresión `assert A.D == B.D` para dos matrices `A` y `B` chequea que los dominios de ambos matrices sean iguales (para la suma, por ejemplo). En caso de que la condición no se cumpla, el programa se caerá y arrojará un error reflejando que las matrices no comparten el dominio. No se probarán casos en los que `assert` incluya una condición falsa.

  Se recomienda rellenar el método en las líneas posteriores a la expresión `assert`.

- Al final del archivo `mat.py`, luego de la línea de símbolos `#`, se define la clase `Mat` propiamente tal, que asume que los métodos anteriores han sido creados. Para efectos de esta tarea, pueden considerar esta parte como una caja negra, que consiste básicamente en tomar los métodos rellenados y definir la clase.

- Es importante que te asegures que tus métodos funcionen correctamente, pues los usarás en la próxima parte del proyecto.

- La gran ventaja de los objetos `Mat` es que la sintaxis para operarlos es como la usual para operar números: por ejemplo, para calcular la suma entre dos objetos `Mat`, digamos A y B, basta con escribir `A + B`, en vez de `add(A, B)`, que es la función que tú programarás y que la clase `Mat` usará solo internamente. Otras operaciones que puedes llevar a cabo utilizando estos objetos son `A - B`, `- A`, `alpha*A` (matriz por escalar), `A == B`, `A[r, c]` (obtener valor en fila `r` y columna `c`), `A[r, c] = alpha` (definir valor en fila `r` y `c`), `A.transpose()`, `v*A` (producto vector-matriz), `A*v` (producto matriz-vector) y `A*B` (producto matriz-matriz).

Dinámica de poblaciones (80 %)

Existen innumerables aplicaciones algorítmicas de uso cotidiano que no podrían existir sin las matrices y el álgebra lineal. Ejemplos notorios incluyen al algorítmo *PageRank* en el cual se basa el buscador de Google, las tan a la moda *Redes Neuronales*, y los modelos de dinámicas de población, que serán el tema de este proyecto.

Como primer acercamiento, se propone analizar un modelo muy simple de la dinámica de una población de conejos. El objetivo es estudiar cómo varía la población en el tiempo. Asumiremos que en el año n, la población total P_n de conejos está compuesta por dos grupos: conejos adultos (representados por x_n) y crías de conejos (representados por y_n), que al año siguiente pasan a ser adultos. Es decir, para cada n, se tiene $P_n = x_n + y_n$. Además, asumiremos que en cada año, cada conejo adulto y cada cría tienen, respectivamente, una probabilidad α y β de sobrevivir hasta el año siguiente. Para el nacimiento de nuevos conejos, utilizaremos un parámetro adicional, γ, para representar la cantidad promedio de crías de conejos nacidos por conejo adulto en un año (ignoraremos el género de los conejos). Por último, asumiremos que las crías nacen al final de cada período, por lo que al período siguiente su número corresponde a γ*(adultos que sobreviven). Con estas hipótesis, tenemos el siguiente sistema de ecuaciones lineales para expresar la población en un año dado, como función de la población del año anterior:

$$\begin{cases} x_{n+1} &= \alpha x_n + \beta y_n \\ y_{n+1} &= \gamma \alpha x_n \end{cases}$$

En base a este modelo, se pide desarrollar los siguientes puntos:

(a) ¿Cuál debería ser el dominio de cada uno de los parámetros α, β y γ ? Explica, en términos cualitativos, cómo sospechas tú que influye el valor de los parámetros en la evolución de la población.

(b) Expresa el sistema de ecuaciones anterior como una ecuación matricial de la forma

$$(x_{n+1}, y_{n+1}) = M * (x_n, y_n).$$

Supongamos que Alicia quiere comenzar un criadero de conejos, para el cual consigue un grupo de 100 crías de conejos (y cero adultos) para comenzar. Es decir, si $n = 0$ es el año de inicio:

$$x_0 = 0, \quad y_0 = 100 \quad \text{y} \quad P_0 = 100.$$

Alicia se pregunta cómo evolucionará su población de conejos durante los próximos 30 años. Ayuda a Alicia realizando simulaciones de distintos escenarios, respondiendo las siguientes preguntas:

(b) Considera primero un escenario en que $\beta = 0{,}25$ (uno de cada cuatro crías sobrevive el primer año y se convierte en adulto) y $\gamma = 4$ (cada conejo adulto tiene un promedio de 4 crías al año). Implementa un algorítmo iterativo para calcular el tamaño del grupo de conejos adultos x_n, el tamaño del grupo de crías y_n, y de la población total P_n, desde $n = 0$ a $n = 30$.

Considera los siguientes casos: i) $\alpha = 0{,}45$, ii) $\alpha = 0{,}5$ y iii) $\alpha = 0{,}55$. Grafica los resultados y describe lo que observas: ¿qué ocurre con la población en cada caso?

(c) Ahora fijaremos $\alpha = 0{,}8$ y $\gamma = 1$. Repite el ejercicio anterior ahora para lo casos: i) $\beta = 0{,}2$, ii) $\beta = 0{,}25$ y iii) $\beta = 0{,}3$. ¿Qué ocurre en cada caso?

(d) Sean ahora $\alpha = 0{,}6$ y $\beta = 0{,}4$. Repite el ejercicio anterior ahora para lo casos: $\gamma = 1{,}5$, $\gamma = 1{,}6667$ y $\gamma = 1{,}75$. ¿Qué ocurre en casa caso?

(e) Encuentre condiciones sobre los parámetros α, β y γ, de forma tal que, a partir de cierto instante:

   (i) La población de conejos se mantenga estable.
   (ii) La población de conejos explote.
   (iii) La población de conejos desaparezca.

(HINT: usa las simulaciones de las partes anteriores (y otras más si es necesario) para primero encontrar una relación entre α, β y γ que asegure la estabilidad de la población, y luego observa lo que ocurre al perturbar esta relación).

(f) Sea M la matriz definida en (b). Deriva una expresión para x_n e y_n utilizando solamente M, n, x_0 e y_0.

(g) A partir del resultado de (f), expresa la cantidad de conejos que tendrá Alicia en 20 años como una operación matricial. Para esto, considera $\alpha = 0{,}5$, $\beta = 0{,}25$ y $\gamma = 4{,}00005$ y calcula el valor de las entradas de la matriz utilizada (los valores numéricos, no en función de los parámetros).

Ahora consideraremos un modelo más realista, para una población de Tortugas Bobas (Caretta caretta), propuesto por Crouse et al. En este ejemplo, la población de tortugas se divide en 7 grupos, definidos según las etapas de desarrollo de estas tortugas (rango etario). El modelo considera, para cada grupo, dos parámetros adicionales: la tasa anual de supervivencia, y la cantidad promedio de huevos puestos por año por individuo. En la siguiente tabla, se muestran los parámetros para cada grupo:

Nº Grupo	Etapa	Rango etario (en años)	Tasa de supervivencia	Huevos puestos
1	huevos, crías	< 1	0,6747	0
2	juveniles pequeños	1-7	0.7857	0
3	juveniles grandes	8-15	0,6758	0
4	subadultos	16-21	0,7425	0
5	criadores novatos	22	0,8091	127
6	remigrantes	23	0,8091	4
7	criadores experimentados	24-54	0,8091	80

Sea $x_{i,n}$ la cantidad de tortugas en etapa i el año n. Considerando las tasas de supervivencia y la cantidad de huevos puestos por año, se puede llegar al siguiente sistema de ecuaciones:

$$\begin{aligned}
x_{1,n+1} &= 127x_{5,n} + 4x_{6,n} + 80x_{7,n} \\
x_{2,n+1} &= 0{,}6747x_{1,n} + 0{,}7370x_{2,n} \\
x_{3,n+1} &= 0{,}0486x_{2,n} + 0{,}6610x_{3,n} \\
x_{4,n+1} &= 0{,}0147x_{3,n} + 0{,}6907x_{4,n} \\
x_{5,n+1} &= 0{,}0518x_{4,n} \\
x_{6,n+1} &= 0{,}8091x_{5,n} \\
x_{7,n+1} &= 0{,}8091x_{6,n} + 0{,}8089x_{7,n}
\end{aligned}$$

Definamos

$$X_n = \begin{pmatrix} x_{1,n} \\ x_{2,n} \\ x_{3,n} \\ x_{4,n} \\ x_{5,n} \\ x_{6,n} \\ x_{7,n} \end{pmatrix}$$

En base a lo anterior:

(i) Encuentra una matriz A tal que $X_{n+1} = AX_n$.

(j) Supongamos que una población de 20 remigrantes llegó a una isla aislada, en la cual no habían torturgas Bobas anteriormente. Grafique el número de tortugas (tanto en cada etapa de desarrollo como el total) que habrán en dicha isla, desde $n = 0$ a $n = 100$. Describa lo que ocurre con la población a través del tiempo.

(k) ¿Cómo cambia la evolución de la población al variar la población inicial de los distintos grupos? ¿Cómo afecta la población inicial al destino de esta especie de tortugas?

(l) Calcula, para distintas poblaciones iniciales, el porcentaje de tortugas en cada una de las etapas de desarrollo en relación a la población total, desde $n = 25$ hasta $n = 100$. Grafica los resultados. ¿Cómo influye la distribución de la población inicial en la distribución de la población en el futuro? Describe lo que observas.

Grupos

Este proyecto deben realizarlo en grupos. Pueden ser **hasta de a 4 integrantes**.

Entrega

El archivo `mat.py` se entrega como archivo `.py`. El desarrollo de las preguntas debe entregarse como un `Jupyter Notebook`. Deben además incluir una "portada": un .pdf con un título, los nombres de los integrantes del grupo y una descripción (resúmen) de lo que están entregando (1 página). La fecha de entrega es el **Lunes 8 de Noviembre hasta las 23:59 hrs**.