Prácticas de SAGE (caldum)
19 de Enero de 2012

1. Introducción

En este ejercicio vamos a ver las posibilidades de programación de SAGE haciendo una implementación del algoritmo de reducción de Gauss de matrices. Es probablemente el algoritmo más conocido por los estudiantes de cualquier curso básico de álgebra lineal. Iremos definiendo una serie de funciones que nos permitirán resolver el problema.

2. Base Teórica

La base teórica de este ejercicio se puede ver en

`http://webs.um.es/leandro`

en el ebook sobre reducción de matrices por filas.

3. Matrices en SAGE

Para definir una matriz en `sage` necesitamos tres datos fundamentales: el cuerpo sobre el que está definida (empezaremos utilizando los números racionales), la dimensión y las entradas que tiene en cada posición.

Por ejemplo, para hacer una matriz de 2 filas y 3 columnas sobre $\mathbb{Q}$ con entradas aleatorias entremos en el entorno sage y escribamos:

```
random_matrix(QQ,2,3)
```

Para introducir las diferentes entradas de la matriz y asignar por ejemplo dicha matriz a la variable A hacemos lo siguiente:

```
A = Matrix(QQ,[[1,2,3],[4,5,6]])
```

Para ver la matriz A, simplemente escribimos A o bien

```
print A.str()
```

Es posible introducir matrices sin especificar el cuerpo, por ejemplo:

```
B = Matrix([[1,2,3],[4,5,6]])
```

producirá un resultado similar al anterior sin imprimimos los resultados. en el segundo caso, sage deduce el cuerpo (o el anillo) base a partir del anillo base de los elementos introducidos. Podemos ver la diferencia poniendo los comandos:

```
A.base_ring()
B.base_ring()
```

Los elementos de una matriz se empiezan a numerar desde 0 y se acceden poniendo la fila y columna entre corchetes:

```
B = Matrix([[1,2,3],[4,5,6]])
B[0,0] == 1
```

Nos debería responder True.

4. Operaciones Elementales

Las operaciones elementales sobre una matriz en sage se realizan del siguiente modo:

1. Intercambio de dos filas: Para intercambiar la fila i y la fija j de una matriz A ponemos

   ```
   A.swap_rows(i,j)
   ```

2. Multiplicar una fila por una constante. Si quieremos multiplicar la fila i por la constante x escribimos

   ```
   A.rescale_row(i,x)
   ```

3. Para sumar a una fila i, la fila j multiplicada por x escribimos

   ```
   A.add_multiple_of_row(i,j,x)
   ```

Ejercicio 4.1. *Entra en el entorno* sage *e introduce la matriz*

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

y realiza las siguientes operaciones elementales el en orden indicado:

1. *Intercambia la fila* 0 *y la fila* 2
2. *Multiplica la fila* 0 *por* $1/7$
3. *Suma a la fila* 3 *la fila* 0 *multiplicada por* -1
4. *Multiplica la fila* 2 *por* -6

Escribe la matriz que resulta de realizar estas operaciones en el cuadernillo de prácticas.

5. Buscando los Elementos Pivote

Para realizar el algoritmo de reducción de Gauss es necesario buscar en cada fila, cual el el primer elemento no nulo (si existe), así como detectar cuando una fila es totalmente nula.

Este primer elemento no nulo lo llamaremos elemento pivote y vamos a hacer una función que nos lo calcule.

Para poder definir esta función necesitamos algunas herramientas básicas:

1. Lazos `for`. Si queremos recorrer un conjunto de valores, podemos hacerlo poniendo `for` e indicando el conjunto sobre el que recorre los valores la variable. Por ejemplo, si queremos ir desde 0 hasta 6 podemo describir este conjunto como `range(7)` (teclea `range(7)` y mira el resultado.

   Si queremos que x recorra todos los valores en este rango escribiríamos

   ```
   for x in range(7):
     print x*x
   ```

   Este ejemplo sencillo nos escribe el cuadrado de todos los valores en este rango.
2. Definición de funciones: Si queremos definir una función f que se le pase un parámetro x y que nos devuelva por ejemplo x^2+1 escribimos

   ```
   def f(x):
     return x*x+1
   ```

Ejercicio 5.1. *Calcula una función que llamarás* `piv` *y que se le pasará como parámetro una matriz A y un índice i que corresponderá a una fila. La función debe devolver -1 si la fila i es nula y si no lo es, nos debe devolver el índice correspondiente a la primera entrada no nula empezando a contar desde* 0. *Realiza el ejercicio primer para matrices con* 3 *columnas y luego utiliza el valor* `A.ncols()` *que te proporciona el número de columnas para hacer una función que resuelva el problema para una matriz cualquiera.*

Escribe en tu cuadernillo de prácticas la función implementada.

6. Normalizando el pivote y haciendo ceros

Una vez detectado cual el el pivote de una fila concreta i, vamos a hacer 1 dicho pivote con una operación elemental y luego eliminar todos los elementos que hay encima y debajo del pivote. Para ello necesitamos comprobar condiciones lógicas y realizar operaciones condicionados a ellas. Eso se hace con `if`. Así, si tenemos que realizar una operación `f(A)` si una cierta variable `p` positiva escribimos

```
if(p>0):
  f(A)
```

En caso de que haya que hacer varias operaciones, como siempre las ponemos todas seguidas respetando la indentación.

Ejercicio 6.1. *Dada una matriz A y una fila i, busca el primer elemento no nulo de dicha fila. Si existe y no es uno, conviértelo en* 1 *multiplicando toda la fila por el valor adecuado utilizando la operación elemental de reescalado.*

Una vez convertido en 1 *el pivote, para cada elemento de la columna distinto de* 0 *y distinto de él mismo, conviértelo en* 0 *sumando a la fila correspondiente, la fila i multiplicada por el negativo del elemento que queremos anular. Cuenta en una variable* `oper` *el número de operaciones reales que realizas y cada vez que hagas una, escribe la matriz resultante.*

Entendemos como operaciones reales aquellas que cambian la matriz, por ejemplo, si reescalamos una fila multiplicándola por 1, *en realidad no cambiamos la matriz, esa no es una operación real, como tampoco lo es sumas a una fila, el contenido de otra multiplicado por* 0.

Introduce la matriz

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

y realiza las operaciones elementales correspondientea a fila 1 *(recuerda que el primer índice es el* 0, *por lo tanto la fila* 1 *es la segunda). Escribe en tu cuadernillo de prácticas el resultado que da la función después de normalizar el pivote de dicha fila y eliminar todos los elementos no nulos de su columna. Escribe también el número de operaciones elementales realizadas.*

7. Terminando el proceso

Para terminar el proceso de reducción y una vez que hemos pasado fila por fila haciendo los pivotes y eliminando todos los elementos no nulos, tenemos que reordenar las filas para poner los pivotes de cada fila más a la derecha que los de la fila anterior y también poner las filas de ceros en la parte inferior.

Para ello tendrás que utilizar un algoritmo de ordenación.

Ejercicio 7.1. *Escribe un programa que dada una matriz, haga su reducida por filas completa escribiendo, cada vez que se realiza una operación elemental, la matriz resultante. Para ello probablemente necesites la función* `A.nrows()` *que te dice el número de filas de la matriz.*

Introduce la matriz

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

y escribe en tu cuadernillo de prácticas la matriz que resulta de realizar exactamente 7 operaciones elementales. Escribe también cuales son estas operaciones.

8. Punto Extra

Ejercicio 8.1. *Estudia los cambios que serían necesarios para que el programa funcionara para matrices sobre un cuerpo arbitrario y realiza pruebas con matrices sobre el cuerpo* GF(5) *y* GF(8,'a') *reduciendo las matrices:*

$$\begin{pmatrix} 2 & 4 & 1 & 3 & 0 \\ 2 & 3 & 0 & 4 & 1 \\ 3 & 4 & 1 & 1 & 4 \\ 1 & 1 & 4 & 3 & 0 \end{pmatrix}$$

$$\begin{pmatrix} a^2+a+1 & a^2+a & 1 \\ a & a & a \\ a^2+1 & 1 & a^2 \end{pmatrix}$$