Manual de Usuario

PROYECTO 4

Ortogonalización de Gram-Schmidt

Mauricio G. Coello | A01328258 | 16 de marzo del 2016

Introducción

El método de ortonormalización de Gram–Schmidt es un algoritmo, que, por medio de un conjunto de vectores linealmente independientes, permite construir otro conjunto ortonormal de vectores que genere el mismo subespacio vectorial.

Manual de Usuario

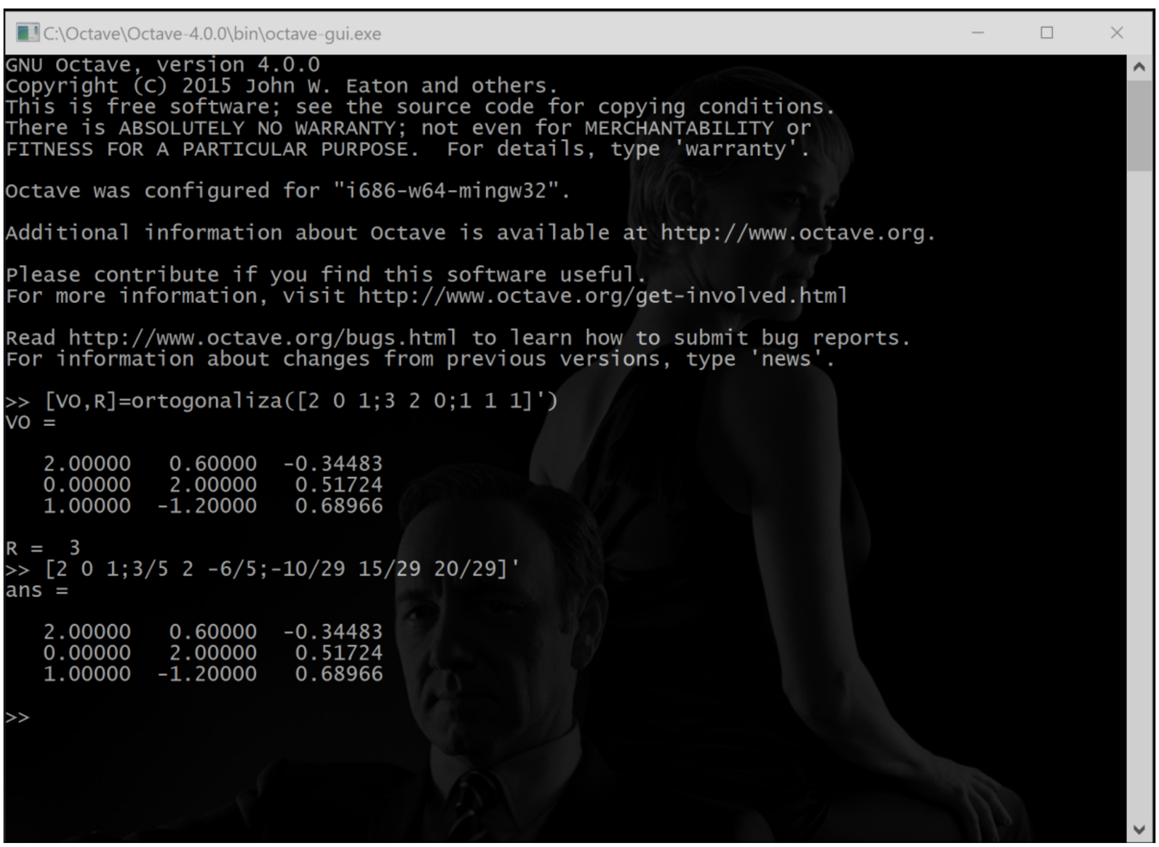
La función principal recibe como parámetros los siguientes valores:

- (V) = Matriz de vectores columna de n dimensiones.
- (eps) = Criterio para determinar si la magnitud es 0.

Por lo que la función debe ser llamada de la siguiente manera

function [VO,R] = ortogonaliza (V,eps)

Ejemplo de funcionalidad



```
C:\Octave\Octave-4.0.0\bin\octave-gui.exe
GNU Octave, version 4.0.0
Copyright (C) 2015 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type 'warranty'.

Octave was configured for "i686-w64-mingw32".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/get-involved.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

>> [VO,R]=ortogonaliza([2 0 1;3 2 0;1 1 1]')
VO =

   2.00000   0.60000  -0.34483
   0.00000   2.00000   0.51724
   1.00000  -1.20000   0.68966

R =  3
>> [2 0 1;3/5 2 -6/5;-10/29 15/29 20/29]'
ans =

   2.00000   0.60000  -0.34483
   0.00000   2.00000   0.51724
   1.00000  -1.20000   0.68966

>>
```

Algoritmo (Gram-Schmidt)

1. Tomar e1=X1
2. $e_2 = X_2 - \alpha_{1,2} e_1$
   $\alpha_{1,2}$debe ser tal que

$$e_1 \cdot e_2 = e$$
$$e_1 \cdot e_2 = 0 = e_1 \cdot (x_2 - \alpha_{1,2} e_1)$$
$$0 = x_2 \cdot e_1 - \alpha_{1,2} e_1 \cdot e_1$$

   Entonces tenemos que

$$\alpha_{1,2} = \frac{x_2 \cdot e_1}{e_1 \cdot e_2}$$

Descripción técnica

La implementación de la función del método Gram-Schmidt utiliza la forma básica del algoritmo original, que dentro de un for loop, hace las respectivas iteraciones.

```
for i=1:x-1
    VO(:, i+1) = V(:, i+1);
    for j=1:i
        a(j, i+1) = (dot(V(:, i+1) , VO(:, j))) / (dot(VO(:, j) , VO(:, j)));
        VO(:, i+1) = VO(:, i+1) - a(j, i+1) * VO(:, j);
    end
end
```

Y en este caso, como nos fue solicitado, en otro for loop, se valida basados es el valor épsilon dado, si la magnitud de uno de los vectores es 0 o no.

```
for j=1:x
    if  norm(VO(:, j)) < eps
        x = x-1;
    else
        tmp(:, j) = VO(:, i);
        i ++;
    end
end
```

Bibliografía

- Presentaciones de clase, Dr. Víctor de la Cueva, 2016
- Nieves, A. & Domínguez, F. (2006). Métodos Numéricos Aplicados a la Ingeniería. México: Continental.