Handin4

March 23, 2022

By Christian Amstrup Petersen, Student number: 202104742

[25]:
```python
import numpy as np

# Resuable functions and variables

# Makes prints of vectors/matrices abit prettier [Purely for aestetics].␣
↪Include name in print if wanted.
def vprint(matrix, name = None):
    if matrix.shape == (4,4):
        print(matrix[2][2])
    pName =  "" if (name == None) else f"{name}"
    if matrix.shape[1] > 1:
        print(f"Matrix {pName}of shape {matrix.shape[0]}x{matrix.shape[1]}:")
    else:
        print(f"Vector {pName}of {matrix.shape[0]} dimensions:")
    for i in range(matrix.shape[0]):
        res = f"\t|  "
        for j in range(matrix.shape[1]):
            res += (f"[{matrix[i][j]}]\t").expandtabs(4)
        print(res+"|")
```

A Gram and the orthogonal vectors

For this assignment we're given the following column matrices:

$$v_0 = \begin{bmatrix} 1.0 \\ -1.0 \\ 1.0 \\ -1.0 \end{bmatrix}, v_1 = \begin{bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{bmatrix}, v_2 = \begin{bmatrix} 2.0 \\ 0.0 \\ -2.0 \\ 0.0 \end{bmatrix}$$

Another thing essential to this assignment is the Grammatrix for three vectors:

$$Grammatrix = \begin{bmatrix} \langle v_0\,, v_0 \rangle & \langle v_0\,, v_1 \rangle & \langle v_0\,, v_2 \rangle \\ \langle v_1\,, v_0 \rangle & \langle v_1\,, v_1 \rangle & \langle v_1\,, v_2 \rangle \\ \langle v_2\,, v_0 \rangle & \langle v_2\,, v_1 \rangle & \langle v_2\,, v_2 \rangle \end{bmatrix}$$

[13]:
```python
# Produces a Gram matrix from the given vectors
def gramMatrix(vectors):
    # Assertion that the amount of vectors > 0
    assert(len(vectors) > 0)

    # Assertion that all given vectors are the same length
    for v in vectors:
        assert(len(vectors[0]) == len(v))
    v = np.hstack(vectors)
    return v.T @ v

# Initializing v0, v1, v2 as column vectors
v0 = np.array([1., -1., 1., -1.], dtype = float)[:, np.newaxis]
v1 = np.array([1., 1., 1., 1.],dtype = float)[:, np.newaxis]
v2 = np.array([2., 0., -2., 0.],dtype = float)[:, np.newaxis]

# 'Pretty' printer for the Grammatrix
vprint(gramMatrix([v0,v1,v2]), "'Grammatrix'")
```

```
Matrix 'Grammatrix'of shape 3x3:
        |  [4.0]   [0.0]   [0.0]   |
        |  [0.0]   [4.0]   [0.0]   |
        |  [0.0]   [0.0]   [8.0]   |
```

We know that if $\langle u\,, v\rangle = 0$, then u and v are orthogonal. Given a set of vectors, that means, that if constructing a Grammatrix from this set, if the result is a diagonal matrix, then the set are orthogonal to each other. This is given by, that the inner product of a vector, with itself, can only be $= 0$, if the vector is a zero-vector, and that if the inner product of two vectors are equal to zero, if the two vectors are orthogonal.

B Projection

Here we want to calculate the projection of the vector x, given by:

$$x = \begin{bmatrix} 3.0 \\ 2.0 \\ 1.0 \\ 0.0 \end{bmatrix},$$

onto our v_0, v_1, and v_2. Here we use definition 8.19, with our u-vector (in this case x), multiplied onto P, so we get:

$$Px = \frac{\langle v_0\,, x\rangle}{||v_0||_2^2} v_0 + \frac{\langle v_1\,, x\rangle}{||v_1||_2^2} v_1 + \frac{\langle v_2\,, x\rangle}{||v_2||_2^2} v_2$$

This rewrite of the definiton is allowed due to:

$$\frac{1}{||v||_2^2} v v^T u = \frac{\langle v\,, u\rangle}{||v||_2^2} v$$

It's important to note that:

$$||v||_2^2 = \langle v\,, v\rangle$$

[41]:
```
def projection(u,v):
    uv = np.vdot(u,v)
    vv = np.vdot(v,v)
    return (uv/vv) * v

# Initializing x as column vector
x = np.array([[3., 2., 1., 0.]],dtype = float).T

pr_x = projection(x,v0) + projection(x,v1) + projection(x,v2)

# 'Pretty' printer for pr_x
vprint(pr_x, "'Px' ")
```

```
Vector 'Px' of 4 dimensions:
        |  [3.0]   |
        |  [1.0]   |
        |  [1.0]   |
        |  [1.0]   |
```

C $x - Px$

For this part, we simply use the same operations, as in a), to determine whether or not they are orthogonal:

[42]:
```
v3 = x - pr_x
# Checking if all orthogonal on v3
for v in [v0, v1, v2]:
    print(np.vdot(v3, v))
```

```
0.0
0.0
0.0
```

D Orthonormal basis

Our main focus in this part of the assignment is the idea of 'orthonormal'. A vector is orthonormal if:

$$\langle v , v \rangle = 1$$

[43]:
```
for v in [v0, v1, v2, v3]:
    print(np.vdot(v,v))
```

```
4.0
4.0
8.0
2.0
```

We see here that our set are not orthonormal. We can convert these by doing the following:

$$v_{norm} = \frac{v}{||v||_2}$$

[37]:
```
v0norm = v0/np.linalg.norm(v0)
v1norm = v1/np.linalg.norm(v1)
v2norm = v2/np.linalg.norm(v2)
v3norm = v3/np.linalg.norm(v3)
```

According to Lemma 9.8, if we take the four normalized vectors above, and use them to calculate a new Grammatrix, and get a I_4-matrix, then we have a orthonormal collection for $\mathbb{R}^4$:

[44]:
```
V = np.hstack((v0norm,v1norm,v2norm,v3norm))
G = V.T @ V
print(G)
```

```
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

Just above **Proposition 9.8**, we get that if our collection is orthonormal in $\mathbb{R}^n$, with n vectors (In our case $n = 4$), we have a orthonormal basis for $\mathbb{R}^n$. In our case, all of our vectors $v_{0...3} \in \mathbb{R}^4$, and we're working with 4 vectors, we can conclude that our collection is a orthonormal basis for $\mathbb{R}^4$.