Daniel Geisler, Daniel Fadin 29.05.2023

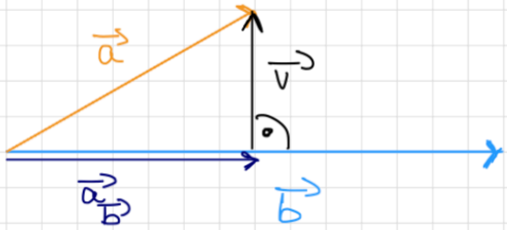
Project: Gram-Schmidt algorithm

In this project, we will go into the basics of the Gram-Schmidt algorithm and also give you a piece of code which computes an orthonormal basis for you, so you don't have to do it anymore (by hand at least).

First, we need to do a short revision on orthogonal projections:

Revision: Orthogonal projection

What is our goal?



We want to adjust $\vec{a}$ in a way that it becomes orthogonal to $\vec{b}$ ($\vec{a} \circ \vec{b} = 0$). For that, we need to find the value of $\vec{a}_{\vec{b}}$ and subtract it from $\vec{a}$.

So we have two conditions:

$$\text{I} \quad \vec{a}_{\vec{b}} = \lambda \cdot \vec{b}$$

$$\text{II} \quad (\vec{a} - \vec{a}_{\vec{b}})^T \circ \vec{b} = 0$$

We have already stated that $\vec{a}_{\vec{b}} + \vec{v} = \vec{a} \Leftrightarrow v = \vec{a} - \vec{a}_b$

We take II and do some transformations:

First, we use I:

$$(\vec{a} - \vec{a}_{\vec{b}})^T \circ \vec{b} = 0$$

$$(\text{I}) \quad (\vec{a} - \lambda \vec{b})^T \circ \vec{b}$$

Then we perform classic multiplications:

$$\vec{a}^T \vec{b} - \lambda \cdot \vec{b}^T \cdot \vec{b}$$

Lastly, we use that $\vec{b}^T \cdot \vec{b} = |b|^2$ and isolate λ:

$$\Rightarrow \lambda = \frac{\vec{a}^T \vec{b}}{|\vec{b}|^2}$$

Now we formulate the final orthogonal projection:

$$\vec{v} = \vec{a} - \frac{\vec{a}^T \vec{b}}{|\vec{b}|^2} \vec{b}$$

Now we use that to proof the Gram-Schmidt-process:

We will proof the Gram-Schmidt process:

Let $\{b_1, \dots, b_n\}$ be a basis, so $\{v_1, \dots, v_n\}$ is an orthonormal basis with $\mathrm{span}(b_1, \dots, b_n)$

(1) $$v_j' = b_j - \sum_{i=1}^{j-1} \langle b_j, v_i \rangle \cdot v_i$$

(2) $$v_j = \frac{1}{\|v_j'\|} v_j' \qquad j = 1 \dots n$$

Proof:

We only show that $\{v_1, \dots, v_n\}$ is an orthogonal basis

all v_j $j = 1 \dots n$ are normalized after construction (2).

induction start:

for $n=1$, $b_1 = v_1'$ $\quad v_1 = \frac{v_1'}{\|v_1'\|}$ $\quad \{v_1\}$ is an orthogonal basis

induction assumption:

For any but fixed n, the above expression applies

inductive step ($n \Rightarrow n+1$)

Let $k \leq n$

$$\langle v_k, v_{n+1} \rangle \overset{(1)}{=} \left\langle v_k, b_{n+1} - \sum_{i=1}^{n} \langle b_{n+1}, v_i \rangle \cdot v_i \right\rangle$$

$$\overset{\text{bilinear}}{=} \langle v_k, b_{n+1} \rangle - \left(\sum_{i=1}^{n} \langle b_{n+1}, v_i \rangle\right) \langle v_k, v_i \rangle$$

$$= \langle v_k, b_{n+1} \rangle - \left(\sum_{i=1}^{n} \langle b_{n+1}, v_i \rangle\right) \cdot \delta_{k,i} \|v_k\|$$

$$= \langle v_k, b_{n+1} \rangle - \langle b_{n+1}, v_k \rangle \|v_k\| \leftarrow v_k \text{ has length } 1$$

$$= \langle v_k, b_{n+1} \rangle - \langle b_{n+1}, v_k \rangle = 0$$

$\Rightarrow$ $\{v_1, \dots, v_{n+1}\}$ is an orthogonal basis

because $v_i \perp v_k$ for $i,k \leq n$, according to the induction assumption

Thus, the formula holds for $n = n+1$ $\square$

Now we translate all of this into code, we start with the orthogonal projection, so the code stays as simple as possible:

```
#
def projection(v,w):
    pV =(v.dot_product(w)/norm(w)^2)*w    #Formula for the orthogonal projection
    return (pV)
#
```

Using our command called „projection“, we can finally code the Gram-Schmidt-process into sage:

```
def Gram_Schmidt(M):
    length = M.nrows()
    E =[ M[0]/ norm(M[0]) ]                                    #We use the first basis vector, normalize it and add it into our final matrix
    for i in range (1 , length ) :                             #We start at 0 in the line before, thats why our algorithm starts at 1
        q = M[i] - sum([ projection(M[i], E[j]) for j in range(i) ]) #Subtract the orthogonal projection of the remaining vectors
        E.append(q / norm(q))                                  #Normalize the orthogonal vector from the step before and add it into the 
matrix

    return E #Return the final matrix
"
```

Applying our code onto the following matrix, gives us the orthonormal vectors which you can validate quickly by hand:

Matrix that has been given: $\begin{pmatrix} 1 & 2 & 5 \\ 1 & 1 & 1 \\ 1 & 0 & 3 \end{pmatrix}$

Calculated matrix: $\begin{pmatrix} \frac{1}{3}\sqrt{3} & \frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6} \\ \frac{1}{3}\sqrt{3} & 0 & -\frac{1}{3}\sqrt{6} \\ \frac{1}{3}\sqrt{3} & -\frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6} \end{pmatrix}$