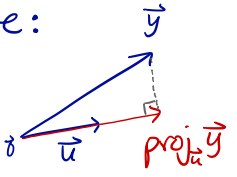
More Orthogonal Projections

28 Apr 2017

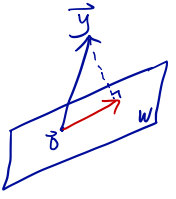
Last time we learned that to project a vector $\vec{y} \in \mathbb{R}^n$ onto the line spanned by $\vec{u}$, we calculate:

$$\text{proj}_{\vec{u}} \vec{y} = \frac{\vec{y} \cdot \vec{u}}{\vec{u} \cdot \vec{u}} \vec{u}$$



To project onto a subspace W of $\mathbb{R}^n$, we take an orthogonal basis $\{\vec{u}_1, \ldots, \vec{u}_p\}$ of W, project onto each $\vec{u}$, & sum the results:

$$(*) \qquad \text{proj}_W \vec{y} = \frac{\vec{y} \cdot \vec{u}_1}{\vec{u}_1 \cdot \vec{u}_1} \vec{u}_1 + \cdots + \frac{\vec{y} \cdot \vec{u}_p}{\vec{u}_p \cdot \vec{u}_p} \vec{u}_p$$



Now, if the basis of W is <u>orthonormal</u>, then $(*)$ reduces to $\vec{u}_1(\vec{y} \cdot \vec{u}_1) + \cdots + \vec{u}_p(\vec{y} \cdot \vec{u}_p)$. If we form the matrix $U = [\vec{u}_1 \cdots \vec{u}_p]$, then the entries of $U^T y$ are $y \cdot \vec{u}_1, \vec{y} \cdot \vec{u}_2, \ldots, \vec{y} \cdot \vec{u}_p$. By definition of the matrix-vector product, then, $\text{proj}_W \vec{y}$ is equal to $UU^T \vec{y}$. So we get:

!! If U is an orthogonal matrix, then $U^T U = I$ and UU^T represents the orthogonal projection onto $\text{Col}\, U$.

Here's a typical example for a 3D subspace in $\mathbb{R}^4$, to give you a sense of how this looks:

In [1]:
```
using SymPy
```

In [2]:
```
srand(123)
A = [Sym[rand(-4:4) for i=1:4] for j=1:3] # random matrix with entries from -4 to 4
hcat(A...)
```

Out[2]:
$$\begin{bmatrix} 0 & 0 & 1 \\ 3 & 4 & -2 \\ 0 & -1 & 4 \\ 4 & -1 & 3 \end{bmatrix}$$

$W = \text{Col}(\text{this matrix})$

In [3]:
```
U = hcat(GramSchmidt(A)...)
U = U ./ sqrt(sum(U.^2,1)) # normalize the columns of U
```

Out[3]:
$$\begin{bmatrix} 0 & 0 & \frac{\sqrt{1492662}}{3867} \\ \frac{3}{5} & \frac{38\sqrt{386}}{965} & \frac{118\sqrt{1492662}}{746331} \\ 0 & -\frac{5\sqrt{386}}{386} & \frac{1121\sqrt{1492662}}{1492662} \\ \frac{4}{5} & -\frac{57\sqrt{386}}{1930} & -\frac{59\sqrt{1492662}}{497554} \end{bmatrix}$$

$W = \text{Col}(\text{this matrix too})$

& this matrix is orthogonal! See!

In [4]:
```
U'*U
```
$= U^T U$

Out[4]:
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In [5]:
```
U*U'
```
$= UU^T$

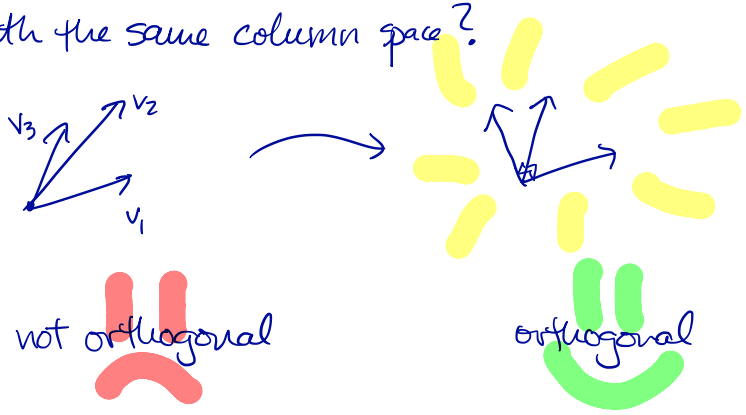
Out[5]:
$$\begin{bmatrix} \frac{386}{3867} & \frac{236}{3867} & \frac{1121}{3867} & -\frac{59}{1289} \\ \frac{236}{3867} & \frac{3851}{3867} & -\frac{76}{3867} & \frac{4}{1289} \\ \frac{1121}{3867} & -\frac{76}{3867} & \frac{3506}{3867} & \frac{19}{1289} \\ -\frac{59}{1289} & \frac{4}{1289} & \frac{19}{1289} & \frac{1286}{1289} \end{bmatrix}$$

multiplying this matrix by any vector projects that vector onto W.

$\text{rank}(UU^T) = 3$

§6.4 Gram-Schmidt

What was that function on the previous page that turned our random matrix into an orthogonal one with the same column space?

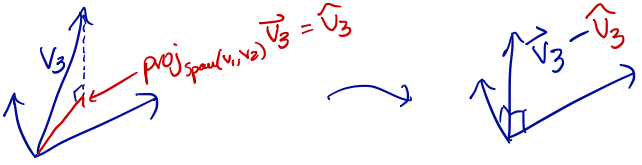


It's called Gram-Schmidt, & the idea is pretty simple:

(1) Leave the first vector as is.

(2) Project the second vector onto the first & subtract that part out to make it orthogonal:

(3) Project the third vector onto the span of the first two; this is easy b/c we already have an orthogonal basis for this span.



④ Etc.

In math notation, suppose $\{\vec{v}_1, \dots, \vec{v}_p\}$ is lin. ind. & set:

(•) $\vec{b}_1 = \vec{v}_1$

(•) $\vec{b}_2 = \vec{v}_2 - \frac{\vec{v}_2 \cdot \vec{v}_1}{\vec{v}_1 \cdot \vec{v}_1} \vec{v}_1$

(•) $\vec{b}_2 = \vec{v}_3 - \frac{\vec{v}_3 \cdot \vec{v}_1}{\vec{v}_1 \cdot \vec{v}_1} \vec{v}_1 - \frac{\vec{v}_3 \cdot \vec{v}_2}{\vec{v}_2 \cdot \vec{v}_2} \vec{v}_2$

$\vdots$

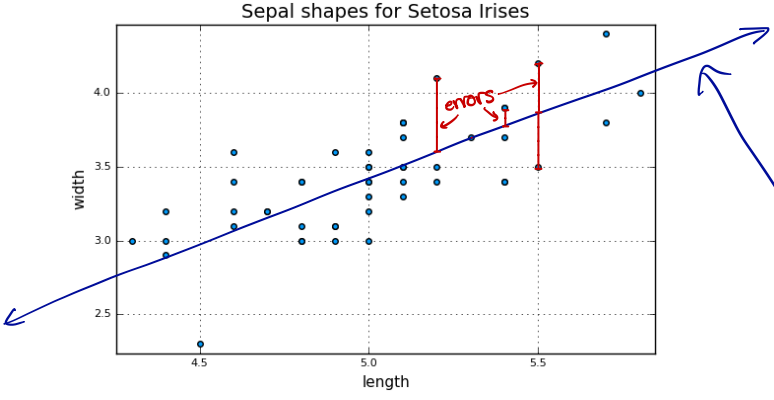
(•) $\vec{b}_p = \vec{v}_p - \frac{\vec{v}_p \cdot \vec{v}_1}{\vec{v}_1 \cdot \vec{v}_1} \vec{v}_1 - \cdots - \frac{v_p \cdot \vec{v}_{p-1}}{\vec{v}_{p-1} \cdot \vec{v}_{p-1}} \vec{v}_{p-1}$

Then $\text{span}\{\vec{b}_1, \dots, \vec{b}_p\} = \text{span}(\{\vec{v}_1, \dots, \vec{v}_p\})$, and $\{\vec{b}_1, \dots, \vec{b}_p\}$ are orthogonal. We can also normalize them to unit length if we want.

§6.5 Least Squares

We will conclude the course by looking at

a central concept in statistics: least squares regression. Suppose we have some data that show a trend:



We'd like to figure out the general slope of a line that roughly follows the data, like this one. One way to do that is to look for the line which minimizes the squared error; the sum of the squares of all the lengths of red segments like the ones shown above. This feels like a calculus problem because we're trying to optimize a

function, but will see that there's an elegant linear algebra solution: define

$\vec{l}$ = the vector of sepal lengths

$\vec{b}$ = the vector of sepal widths

Then what we want is to find $\alpha, \beta \in \mathbb{R}$ so that

$$\left\| \vec{b} - \left(\alpha \vec{1} + \beta \vec{l} \right) \right\|^2$$

($\vec{1}$: vector of all ones)

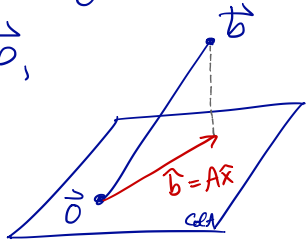
is as small as possible. Letting $A = [\vec{1} \ \vec{l}]$ and $\vec{x} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, we're looking to minimize

$$\left| A\vec{x} - \vec{b} \right|^2.$$

This we know how to do! $A\vec{x} \in \text{Col } A$, so we're looking for the vector in Col A that is as close to $\vec{b}$ as possible. The orthogonal projection of $\vec{b}$ onto Col A!

[Note: Col A is a 2D subspace in $\mathbb{R}^n$ where n is the number of data points, thus typically quite large.]

Let's figure out how to solve this equation without orthogonalizing. Let $\hat{b} = \text{proj}_{\text{Col}A}\vec{b}$, & let $\hat{x}$ satisfy $A\hat{x} = \hat{b}$. Then each column of A will be orthogonal to $\vec{b} - \hat{b}$: Let $A = [\vec{a}_1 \cdots \vec{a}_n]$; then



$$\vec{a}_j \cdot (\vec{b} - \hat{b}) = 0 \quad \text{for all } j \Rightarrow$$

$$\vec{a}_j^T (\vec{b} - \hat{b}) = 0 \quad \text{for all } j \Rightarrow$$

$$A^T(\vec{b} - \hat{b}) = \vec{0}.$$

Now this equation says $A^Tb = A^TA\hat{x}$. If A^TA is invertible, then $\hat{x} = (A^TA)^{-1}A^T\vec{b}$ is the least squares solution! This is a very famous formula you're sure to see again.

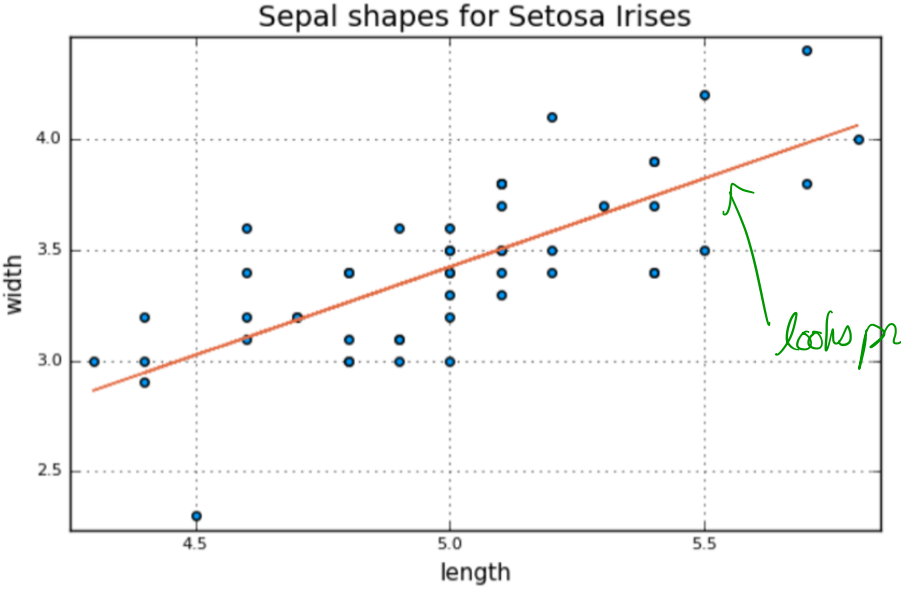
Let's see how this works: [next page]

We load the data, calculate $(A^TA)^{-1}A^T\vec{b}$ directly, & plot the results.

this is the important line: $(A^TA)^{-1}A^T\vec{b}$

```
using Plots
using RDatasets
iris = dataset("datasets","iris");
setosas = iris[:Species] .== "setosa";
l = iris[:SepalLength][setosas] # vector of lengths
b = iris[:SepalWidth][setosas] # vector of widths
A = [ones(length(l)) l]
α, β = inv(A'*A)*A'*b
p = scatter(iris[:SepalLength][setosas],iris[:SepalWidth][setosas],
xlabel="length",ylabel="width",legend=nothing,title="Sepal shapes for Setosa Irises")
plot!(p,l,α + l*β)
```

← line plot



looks pretty good!