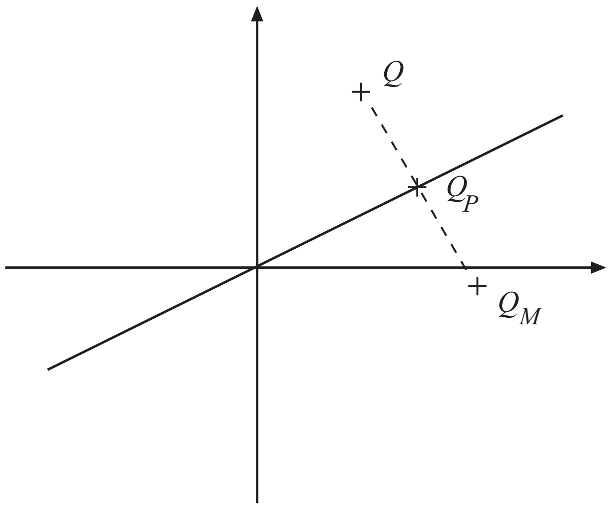
Holger Broman, August 19, 1999

Projections

Rather than repeating the intuitive build-up as done in the section on reflections, let us sit back and do a little "high-level reasoning". What is the relation between an original point, its mirror image, and its projection onto the same subspace as that of the reflection? Yes, of course, and let us formalize. The vector to the original point is $\mathbf{x}$, to its mirror image $\mathbf{x}_M$ and to its projection $\mathbf{x}_P$. Thus,

$$\mathbf{x}_P = \frac{\mathbf{x} + \mathbf{x}_M}{2},$$

as shown in the figure below.



As projecting a point is a linear mapping, there must exist a projection matrix. Denote it by P and the reflection matrix by M. The relation above then reads

$$\mathbf{x}_P = P\mathbf{x} = \frac{1}{2}(\mathbf{x} + M\mathbf{x}) = \frac{1}{2}(M + I)\mathbf{x}.$$

Re-write

$$\left[P - \frac{1}{2}(M + I)\right]\mathbf{x} = 0.$$

As this relation holds true for all choices of $\mathbf{x}$, we draw the conclusion that the range space of $P - \frac{1}{2}(M + I)$ must be empty and that

$$P = \frac{1}{2}(M + I).$$

We can use this relation to derive properties of P:

- M symmetric $\Rightarrow$ P symmetric
- $I = M^2 = (2P - I)^2 = 4P^2 - 4P + I \Rightarrow P^2 = P$

The second property we could have derived without doing any calculations: projecting an already projected point does not move it any further.

Now, we are again prepared to give a sensible definition:

Definition: A symmetric matrix P is an orthogonal projection matrix iff $P^2 = P$.

Note 1 Projection matrices are very important not only in Least Squares estimation problems, but also in other areas of modern signal processing.

Note 2 The relation between reflections and projections has been stated, a relation that makes it possible to construct one matrix from the other.

———— o O o ————

The eigenvalues of projection matrices are easily found:

I

$$P\mathbf{g} \triangleq \lambda\mathbf{g}$$
$$\lambda\mathbf{g} = P\mathbf{g} = P^2\mathbf{g} = \lambda^2\mathbf{g}$$

Thus,

$$\lambda \in \{0, 1\}$$

II

$$0 = \det(P - \lambda_P I) = \det\left[\frac{1}{2}(M + I) - \lambda_P I\right] =$$
$$= \det\left[\frac{1}{2}(M + I - 2\lambda_P I)\right] \sim \det\Big[M - (2\lambda_P - 1)I\Big]$$

Thus,

$$\lambda_M = 2\lambda_P - 1; \qquad \lambda_P = \frac{\lambda_M + 1}{2}$$

How many of the eigenvalues of P equal 1? Well, the rank of P equals not only this number, but also the dimension of the range space of P. The range space of P is of course the space onto which P projects.

———— o O o ————

How do you construct the projection matrix given the space onto which to project? The answer will be given taking the (imaginary) detour of solving overdetermined systems of linear equations.

Given an overdetermined system of equations

$$A\mathbf{x} = \mathbf{b},$$

you will know from the next leaflet that the LS solution is given by

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b},$$

where we assumed that $A^T A$ is non-singular. Thus, the equation

$$A\mathbf{x} = A(A^T A)^{-1} A^T \mathbf{b}$$

is guaranteed to have exactly one solution. How come, when the equation

$$A\mathbf{x} = \mathbf{b}$$

generally has no solution? The answer is that the vector $A(A^T A)^{-1} A^T \mathbf{b}$ must belong to the range space of A – remember the important interpretation of $A\mathbf{x}$ as a linear combination of the column vectors of A. Could it be that the matrix $A(A^T A)^{-1} A^T$ is a projection matrix?

- $\left[A(A^T A)^{-1} A^T\right]^T = \ldots = A(A^T A)^{-1} A^T$
- $\left[A(A^T A)^{-1} A^T\right]^2 = \ldots = A(A^T A)^{-1} A^T$

Yes!

- A vector that falls in the range space of A can be written as a linear combination of the columns of A, i.e. as $A\mathbf{y}$. What does the projection matrix do to this vector?

$$A(A^T A)^{-1} A^T A\mathbf{y} = A\mathbf{y}$$

Thus, $P = A(A^T A)^{-1} A^T$ projects onto the space spanned by the columns of A. We have thus proved the following:

Theorem: Given $\mathbf{v}_1, \ldots, \mathbf{v}_M$ that span an M-dimensional subspace of $\mathbb{R}^N$, construct the matrix

$$A = (\mathbf{v}_1, \ldots, \mathbf{v}_M).$$

Then the projection matrix

$$P = A(A^T A)^{-1} A^T$$

projects onto the subspace.

Note 3 Now that we know how to construct projection matrices, we also know how to construct reflection matrices:

$$M = 2P - I$$

Note 4 In practice, you do not construct projection matrices from the formulas

$$\begin{aligned} A &= (\mathbf{v}_1 \ \cdots \ \mathbf{v}_M) \\ P &= A \left(A^T A\right)^{-1} A^T, \end{aligned}$$

it is numerically not a smart thing to do. In the leaflet on orthogonalization, you will learn how. Let us mention how desirable it is to have the vectors $\mathbf{v}_m$ mutually orthonormal, as $A^T A$ then equals the identity and

$$P = A\, A^T,$$

so there is no need to invert any matrix.

———— o O o ————

Let us look at a matrix that is strongly related to the permutation matrix P derived above, i.e. the matrix that projects onto the subspace spanned by $\mathbf{v}_1, \ldots, \mathbf{v}_M$. First, remember that $P\mathbf{x}$ is the projection of $\mathbf{x}$ onto that subspace. The “rest” of $\mathbf{x}$ is

$$\mathbf{y} = \mathbf{x} - P\mathbf{x} = (I - P)\mathbf{x}.$$

Observe that $\mathbf{y}$ is perpendicular to the subspace onto which P projects:

$$P\mathbf{y} = P(I - P)\mathbf{x} = \left(P - P^2\right)\mathbf{x} = 0,$$

so that $\mathbf{y}$ is perpendicular to $\mathbf{v}_1, \ldots, \mathbf{v}_M$.

Study

$$(I - P)^2 = I - 2P + P^2 = I - P,$$

so that $I - P$ also is a projection matrix. This matrix projects onto the orthogonal complement of the subspace spanned by $\mathbf{v}_1, \ldots, \mathbf{v}_M$ and is denoted $P^\perp$:

$$P^\perp = I - P$$

The notation $P^\perp$ is read “P-perp”. $P^\perp$ will play an important role in the procedure to orthogonalize a given set of vectors.

Please run the m-file projections in Matlab.

```
% projections.m, 3D to 2D
% run this program several times, random data.
clear

% Generate the space on which to project
v1=randn(3,1); v2=randn(3,1); A=[v1 v2];

% Construct the projection matrix and check for key properties
P=A*inv((A'*A))*A';
norm(P-P','fro'), norm(P*P-P,'fro')

% Create an ON-basis for the plane and generate plane-plot
[q,r]=qr(A,0);
step=0.2; width=3; index=1;
for l=-width:step:width
p1=l*q(:,1)-width*q(:,2);
p2=l*q(:,1)+width*q(:,2);
p3=(l+step/2)*q(:,1)+width*q(:,2);
p4=(l+step/2)*q(:,1)-width*q(:,2);
planeplot(:,index)=p1;
planeplot(:,index+1)=p2;
planeplot(:,index+2)=p3;
planeplot(:,index+3)=p4;
index=index+4;
end
ax=[ -width width -width width -width width];

% Generate some random vectors and project them
M=50;
dum=randn(3,M);
dummer=P*dum;

% plot some examples
figure(1), clf, hold on, axis(ax), axis equal, view(3)
plot3([0 v1(1)], [0 v1(2)], [0 v1(3)], 'b')
plot3([0 v2(1)], [0 v2(2)], [0 v2(3)], 'b')
for l=1:5
plot3(dum(1,l),dum(2,l),dum(3,l),'rp')
plot3(dummer(1,l),dummer(2,l),dummer(3,l),'gp')
plot3([dum(1,l) dummer(1,l)], [dum(2,l) dummer(2,l)],...
 [dum(3,l) dummer(3,l)], 'y')
end
plot3(planeplot(1,:), planeplot(2,:), planeplot(3,:), 'k')
title('Five points and their projections')

% plot all projected points
figure(2), clf, hold on, axis(ax), axis equal, view(3)
```

```
plot3([0 v1(1)], [0 v1(2)], [0 v1(3)], 'b')
plot3([0 v2(1)], [0 v2(2)], [0 v2(3)], 'b')
for l=1:M
plot3(dummer(1,l),dummer(2,l),dummer(3,l),'gp')
end
plot3(planeplot(1,:), planeplot(2,:), planeplot(3,:), 'k')
title('Fifty projected points')
```