Computer Lab 8: Orthogonal projections and Lines of Best Fit

8.1 Introduction

This lab will investigate an important and useful applications of projections: computing least squares to calculate lines of best fit to model data. The least squares method is a commonly used type of linear regression which is widely used in statistics.

Before starting the exercises, load this week's m-files into the Current Folder in MATLAB by following the instructions on the Computer Lab Class Files page of the MAST10007 website.

8.2 Exercises

Exercise 1: Projection in $\mathbb{R}^2$ and $\mathbb{R}^3$

Given two non-zero vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, the *orthogonal projection* of $\mathbf{v}$ onto $\mathbf{u}$ is

$$\text{proj}_{\mathbf{u}}\,\mathbf{v} = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\|^2}\mathbf{u} = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle}\mathbf{u}$$

The projection is the closest vector to $\mathbf{v}$ that is parallel to the vector $\mathbf{u}$. The vector $\mathbf{v} - \text{proj}_{\mathbf{u}}\,\mathbf{v}$ is perpendicular to $\mathbf{u}$, and is sometimes called the *complementary projection*.

We're going to create an m-file that calculates the orthogonal projection of $\mathbf{v}$ onto $\mathbf{u}$.

To refresh your memory of what the code for an m-file looks like, find the file rowadd2.m in the Current Folder pane, and double-click on it. The file will open in a new pane called Editor.

Recall that the first line names the function and establishes its inputs and outputs. Any line that starts with % is a comment that won't be executed by MATLAB.

Try writing an m-file that calculates the orthogonal projection of $\mathbf{v}$ onto $\mathbf{u}$, using MATLAB's `dot` function. Remember to include all the defining information for the m-file in its first line.

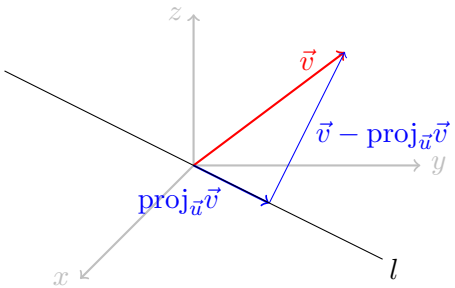
```
function w = proj(v,u)
    % this function projects the vector v onto u
```

Once you've written your function, save it as `proj.m`, and make sure it's in the Current Folder pane in MATLAB. Choose values for two vectors $\mathbf{u}$ and $\mathbf{v}$ and try using `proj` on them. Did your code work as expected? If not, try debugging it until it does.

In particular, what happens if $\mathbf{u} = \mathbf{0}$? Does it make sense geometrically to project a vector onto the zero vector?

Using your `proj` function, what is the closest point to $\mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ on the line ℓ parallel to $\mathbf{u} = \begin{bmatrix} 5 \\ 12 \\ -7 \end{bmatrix}$?

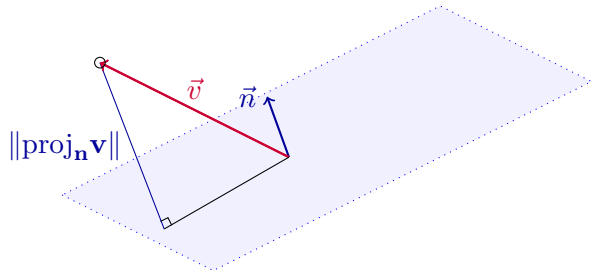
Hence what is the distance between P and ℓ? You might find the picture below helpful.



Note: You can calculate the length of a vector in MATLAB using the function `norm(v)`

We can also use our `proj` function to calculate the distance between a *plane* and a point in $\mathbb{R}^3$, by performing orthogonal projection onto a normal vector.

Consider the plane Π with vector equation $\mathbf{r} = s \begin{bmatrix} 5 \\ -1 \\ 1 \end{bmatrix} + t \begin{bmatrix} 4 \\ -3 \\ 0 \end{bmatrix}$, $s, t \in \mathbb{R}$.



What is a normal vector $\mathbf{n}$ for the plane? Recall, a vector $\mathbf{n}$ is said to be a normal to the plane, if it is perpendicular to the plane, that is for all $\mathbf{r} \in \Pi$, $\langle \mathbf{r}, \mathbf{n} \rangle = 0$.

Calculate the projection of $\mathbf{v} = \begin{bmatrix} 5.5 \\ 4 \\ 7 \end{bmatrix}$ onto $\mathbf{n}$ using `proj` and the distance between $\mathbf{v}$ and Π

$\text{proj}_{\mathbf{n}}\,\mathbf{v} =$ distance $= \|\,\text{proj}_{\mathbf{n}}\,\mathbf{v}\| =$

Recall that taking the orthogonal projection of $\mathbf{v}$ onto $\mathbf{n}$ splits $\mathbf{v}$ into two components; one which is parallel to $\mathbf{n}$, and the other which is perpendicular to $\mathbf{n}$ (and hence parallel to Π).

Given that, what is the orthogonal projection of $\mathbf{v}$ onto Π?

Let A be the matrix with columns given by the vectors $\begin{bmatrix} 5 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 4 \\ -3 \\ 0 \end{bmatrix}$, and let P be the corresponding projection matrix $P = A(A^TA)^{-1}A^T$. Fact check your answer by calculating $P\mathbf{v}$.

Exercise 2: Sampling and lines of best fit

As we have seen, the projection of a vector $\mathbf{v}$ onto a subspace $W \subseteq V$ can be understood as finding the closest point $\mathbf{p} \in W$ to $\mathbf{v}$, i.e. the point which minimises $\|\mathbf{v} - \mathbf{p}\|$. One possible application of this is in calculating lines which best fit a set of data points. In general, given a set of points

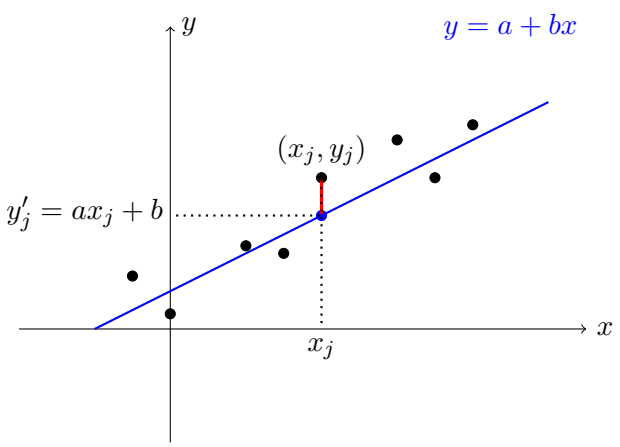
$$\left\{ \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \ldots, \begin{bmatrix} x_n \\ y_n \end{bmatrix} \right\}$$

it won't be possible to find a straight line that will pass through all points. This would only be possible if the data points happen to be collinear, which is rare in empirical sciences for various reasons (not least of which are measurement errors). Fortunately, the method of least squares allows us to calculate a new set of points

$$\left\{ \begin{bmatrix} x_1 \\ y_1' \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2' \end{bmatrix}, \ldots, \begin{bmatrix} x_n \\ y_n' \end{bmatrix} \right\}$$

which *are* collinear, and which minimise $\sum_{i=1}^{n}(y_i' - y_i)^2$, so that the line will be a reasonably good approximation of the data. Stated another way, we are trying to find the polynomial $a + bx$ which minimises the value of

$$\|\mathbf{y} - Sample(a + bx)\|.$$



This is where projections enter the picture. Let

$$\begin{aligned} Sample : \mathcal{P}_1 &\longrightarrow \mathbb{R}^7 \\ \boldsymbol{p} &\longmapsto \begin{bmatrix} p(-3) \\ p(-2) \\ p(-1) \\ p(0) \\ p(1) \\ p(2) \\ p(3) \end{bmatrix} \end{aligned}$$

be the sampling linear transformation with seven sampling points $x = -3$, -2, -1, 0, 1, 2, and 3.

We will start by computing the standard matrix representation of S.

By hand, compute $Sample(\mathbf{1})$ and $Sample(\mathbf{x})$.

$Sample(\mathbf{1}) =$ $Sample(\mathbf{x}) =$

The matrix representation of *Sample* with respect to the standard bases of $\mathcal{P}_1$ and $\mathbb{R}^7$ has columns given by these two vectors. Enter this matrix into MATLAB as a matrix `A`.

Test your matrix is correct by computing `A*[3;7]` and comparing to $Sample(3 + 7x)$.

Suppose we have measured some real world data points, for example:

x	-3	-2	-1	0	1	2	3
y	-0.6	0.2	1	1	1.4	2.3	2.3

Enter $\mathbf{x} = (-3, -2, -1, 0, 1, 2, 3)^T$ and $\mathbf{y} = (-0.6, 0.2, 1, 1, 1.4, 2.3, 2.3)^T$ into MATLAB.

As we've seen in a previous lab, we can plot these points using MATLAB's `scatter` command. Enter the command

```
»scatter(x,y,'r')
```

The third argument in this function controls the colour of the data points: `'r'` will draw the points in red. Don't close this graphics window yet, as we will need it later.

Do all the points lie on a straight line?

By calculating the reduced row-echelon form of an appropriate matrix in MATLAB, show that the linear system

$$A\begin{bmatrix} a \\ b \end{bmatrix} = \mathbf{y}$$

is inconsistent.

Is $\mathbf{y}$ in the column space of A?

Now, let's modify our data set to be collinear.

Using the formula from lectures, calculate the matrix P for projection onto the column space of A. Then, use your answer to calculate $P\mathbf{y}$.

$P\mathbf{y} =$

The vector $P\mathbf{y}$ is the closest point in the column space of A to the real data points $\mathbf{y}$.

To plot these data points on the same set of axes as before, type the following commands:

```
»hold on

»scatter(x,P*y,'b')
```

The points should be drawn in blue. We now want to find the corresponding linear polynomial which fits these points.

Compute the vector $\begin{bmatrix} a \\ b \end{bmatrix} = (A^TA)^{-1}A^T\mathbf{y}$.

Deduce that $A\begin{bmatrix} a \\ b \end{bmatrix} = P\mathbf{y}$, and hence that $P\mathbf{y}$ is in the column space of A.

Since A was the matrix representation of *Sample*, what this shows is that

$$Sample(a + bx) = P\mathbf{y}.$$

In other words, the line $y = a + bx$ fits the data points given by the vector $P\mathbf{y}$, so the points must be collinear. We can confirm this using MATLAB's `fplot` command.

Run the command

```
»fplot(@(x)a+b*x,[-3,3])
```

where `a` and `b` are the values you found above.

Does the line pass through the blue data points?

It is also possible to use the same techniques to approximate data points with other more interesting functions such as sound waves. This is the idea underpinning Fourier analysis, in which we approximate our points by linear combinations of wave functions:

$$\mathcal{W} = \{1, \cos(x), \sin(x), \cos(2x), \sin(2x), \cos(3x), \sin(3x), ...\} \subseteq \mathcal{F}(\mathbb{R}, \mathbb{R}).$$

Let U be the subspace consisting of the span of just the first three basis vectors, i.e. $U = \text{span}\{1, \cos(x), \sin(x)\}$ and consider the linear transformation $Sample : U \to \mathbb{R}^3$, with the same data vectors $\mathbf{x}, \mathbf{y}$ as before.

Compute $Sample(1)$, $Sample(\cos(x))$ and $Sample(\sin(x))$, and hence calculate the matrix representation of $Sample : U \to \mathbb{R}^3$. Enter this into MATLAB as a matrix B.

Calculate $\begin{bmatrix} c \\ d \\ e \end{bmatrix} = (B^T B)^{-1} B^T \mathbf{y}$.

This gives us the coefficients of our three basis vectors for the space U. Plot the corresponding wave function to the graphics window, with the command

```
»fplot(@(x)c+d*cos(x)+e*sin(x),[-3,3])
```

We could achieve a better approximation by taking more than three wave functions from the linearly independent set $\mathcal{W}$, thus replacing U by a higher dimensional wave function space. For example, you could try taking the first five wave functions, instead of the first three. However, one should not make the dimension of U higher than the number of sampling points! This runs the risk of 'overfitting' our approximation, where solutions exist but are no longer unique.