In [9]:

```python
import numpy as np
import torch as t
from torch.distributions import MultivariateNormal as MvNormal
```

Question sheet 1: maximum likelihood regression

Question 1

Derive the regularised maximum likelihood solution to the following optimization problem,

$$\mathcal{L}(\mathbf{w}) = \log \mathrm{P}\left(\mathbf{y}|\mathbf{X},\mathbf{w}\right) - \tfrac{1}{2}\mathbf{w}^T \boldsymbol{\Lambda} \mathbf{w} \tag{1}$$

Answer

We begin by taking the gradient of $\log \mathrm{P}\left(\mathbf{y}|\mathbf{X},\mathbf{w}\right)$ from the notes,

$$\frac{\partial \log \mathrm{P}\left(\mathbf{y}|\mathbf{X},\mathbf{w}\right)}{\partial \mathbf{w}} = \tfrac{1}{\sigma^2}\mathbf{X}^T\left(\mathbf{y} - \mathbf{X}\mathbf{w}\right) \tag{2}$$

Next, we consider the gradient of the second term,

$$\begin{align}
\frac{\partial}{\partial w_\alpha}\left[-\tfrac{1}{2}\sum_{ij} w_i \Lambda_{ij} w_j\right] &= -\tfrac{1}{2}\left[\sum_{ij}\frac{\partial w_i}{\partial w_\alpha}\Lambda_{ij} w_j + \sum_{ij} w_i \Lambda_{ij}\frac{\partial w_j}{\partial w_\alpha}\right] \tag{3}\\
&= -\tfrac{1}{2}\left[\sum_j \Lambda_{\alpha j} w_j + \sum_i w_i \Lambda_{i\alpha}\right] \tag{4}
\end{align}$$

as $\boldsymbol{\Lambda}$ is symmetric,

$$\begin{align}
&= -\tfrac{1}{2}\left[\sum_j \Lambda_{\alpha j} w_j + \sum_i w_i \Lambda_{\alpha i}\right] \tag{5}\\
&= -\sum_i \Lambda_{\alpha i} w_i \tag{6}
\end{align}$$

Putting everything back in matrix notation,

$$\frac{\partial}{\partial \mathbf{w}}\left[-\tfrac{1}{2}\mathbf{w}^T\boldsymbol{\Lambda}\mathbf{w}\right] = -\boldsymbol{\Lambda}\mathbf{w} \tag{7}$$

Combining the first and second terms, we can compute the gradient of the objective,

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = \tfrac{1}{\sigma^2}\mathbf{X}^T\left(\mathbf{y} - \mathbf{X}\mathbf{w}\right) - \boldsymbol{\Lambda}\mathbf{w}. \tag{8}$$

Finally, we solve for the location, $\hat{\mathbf{w}}$, where this gradient is zero,

$$\mathbf{0} = \mathbf{X}^T \left(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}\right) - \mathbf{\Lambda}\hat{\mathbf{w}} \tag{9}$$

$$\mathbf{0} = \mathbf{X}^T \left(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}}\right) - \sigma^2\mathbf{\Lambda}\hat{\mathbf{w}} \tag{10}$$

$$\mathbf{0} = \mathbf{X}^T \mathbf{y} - \mathbf{X}^T\mathbf{X}\hat{\mathbf{w}} - \sigma^2\mathbf{\Lambda}\hat{\mathbf{w}} \tag{11}$$

$$\mathbf{0} = \mathbf{X}^T \mathbf{y} - \left(\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{\Lambda}\right)\hat{\mathbf{w}} \tag{12}$$

$$\left(\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{\Lambda}\right)\hat{\mathbf{w}} = \mathbf{X}^T \mathbf{y} \tag{13}$$

$$\hat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{\Lambda}\right)^{-1}\mathbf{X}^T \mathbf{y} \tag{14}$$

Question 2

For the data sample in the table, and a model of the form $y = w_0 + w_1 x$, a noise-level of $\sigma = 1$, and a regulariser, $\mathbf{\Lambda} = 2\mathbf{I}$, compute the regularised ML solution.

$$\mathcal{L}\left(\mathbf{w}\right) = \log \mathcal{N}\left(\mathbf{y}; \mathbf{X}\mathbf{w}, \sigma^2\right) - \tfrac{1}{2}\mathbf{w}^T\mathbf{\Lambda}\mathbf{w} \tag{15}$$

```
\begin{tabular}{rr}
 x  & y  \\
 \hline
  -2.0 &  -6.2 \\
  -1.0 &  -2.6 \\
   0.0 &   0.5 \\
   1.0 &   2.7 \\
   2.0 &   5.7
\end{tabular}
```

Do this using a calculator, as if you were in an exam.

Answer

First, write down $\mathbf{X}$, $\mathbf{y}$, $\mathbf{\Lambda}$ and σ for error-checking

In [3]:
```
X = t.tensor([
    [1., -2.],
    [1., -1.],
    [1.,  0.],
    [1.,  1.],
    [1.,  2.]
])

y = t.tensor([
    [-6.2],
    [-2.6],
    [ 0.5],
    [ 2.7],
    [ 5.7]
])

La = 2*t.eye(2)
s2 = 1
```

Begin by computing $\mathbf{X}^T\mathbf{X}$,

In [4]:
```
XTX = t.zeros(2,2)

XTX[0,0] = ( 1.)**2 + ( 1.)**2 + ( 1.)**2 + ( 1.)**2 + ( 1.)**2
XTX[1,1] = (-2.)**2 + (-1.)**2 + ( 0.)**2 + ( 1.)**2 + ( 2.)**2
```

```
XTX[0,1] = 1.*(-2.) + 1.*(-1.) + 1.*( 0.) + 1.*( 1.) + 1.*( 2.)
XTX[1,0] = XTX[0,1]

assert t.allclose(XTX, X.T@X)
XTX
```

Out[4]:
```
tensor([[ 5.,  0.],
        [ 0., 10.]])
```

Next compute, $\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{\Lambda}$,

In [5]:
```
XTX_s2La = t.zeros(2,2)

XTX_s2La[0,0] = XTX[0,0] + s2*2
XTX_s2La[1,1] = XTX[1,1] + s2*2
XTX_s2La[1,0] = XTX[1,0]
XTX_s2La[0,1] = XTX[0,1]

assert t.allclose(XTX_s2La, X.T@X + s2*La)
XTX_s2La
```

Out[5]:
```
tensor([[ 7.,  0.],
        [ 0., 12.]])
```

Now, compute $\left(\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{\Lambda}\right)^{-1}$ inverse using the standard formula,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{16}$$

In [6]:
```
inv_XTX_s2La = t.zeros(2,2)

det = XTX_s2La[0,0]*XTX_s2La[1,1] - XTX_s2La[1,0]*XTX_s2La[0,1]
print(det)

inv_XTX_s2La[0,0] =  XTX_s2La[1,1]/det
inv_XTX_s2La[1,1] =  XTX_s2La[0,0]/det
inv_XTX_s2La[1,0] = -XTX_s2La[1,0]/det
inv_XTX_s2La[0,1] = -XTX_s2La[0,1]/det

assert t.allclose(inv_XTX_s2La, t.inverse(X.T@X + s2*La))
inv_XTX_s2La
```

```
tensor(84.)
```

Out[6]:
```
tensor([[0.1429, -0.0000],
        [-0.0000, 0.0833]])
```

Now, compute $\mathbf{X}^T\mathbf{y}$,

In [7]:
```
XTy = t.zeros(2, 1)

XTy[0,0] = ( 1.)*(-6.2) + ( 1.)*(-2.6) + ( 1.)*(0.5) + ( 1.)*(2.7) + ( 1.)*(5.7)
XTy[1,0] = (-2.)*(-6.2) + (-1.)*(-2.6) + ( 0.)*(0.5) + ( 1.)*(2.7) + ( 2.)*(5.7)

assert t.allclose(XTy, X.T@y)
XTy
```

Out[7]:
```
tensor([[ 0.1000],
        [29.1000]])
```

Finally, we compute $\left(\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{\Lambda}\right)^{-1}\mathbf{X}^T\mathbf{y}$ as a matrix-vector multiplication,

In [8]:
```
wh = t.zeros(2, 1)
```

```python
wh[0,0] = inv_XTX_s2La[0,0] * XTy[0,0] + inv_XTX_s2La[0,1] * XTy[1,0]
wh[1,0] = inv_XTX_s2La[1,0] * XTy[0,0] + inv_XTX_s2La[1,1] * XTy[1,0]

assert t.allclose(wh, t.inverse(X.T@X + s2*La) @ X.T@y)
wh
```

Out[8]:

```
tensor([[0.0143],
        [2.4250]])
```