MAS212 Scientific Computing and Simulation

#7: Fitting Models to Data

Key resources:

- Lec 7: `http://sam-dolan.staff.shef.ac.uk/mas212/docs/l7.pdf`
- Data sets : `http://sam-dolan.staff.shef.ac.uk/mas212/data/`.

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline   # include plots in ipython notebook
```

---

1. A straight-line fit. Download the data set (`ds1.txt`), open with `np.loadtxt()`, and plot:

```
x,y = np.loadtxt('ds1.txt')
plt.plot(x, y, 's')
```

(a) Using `numpy` (or writing your own functions) calculate the means $\bar{x}$ and $\bar{y}$, the variance $\mathrm{var}(x) = \frac{1}{N}\sum(x_i - \bar{x})^2$ and the covariance, $\mathrm{covar}(x,y) = \frac{1}{N}\sum(x_i - \bar{x})(y_i - \bar{y})$. Hence find the parameters of the line of best fit:

$$y = mx + c, \qquad m = \frac{\mathrm{covar}(x,y)}{\mathrm{var}(x)}, \quad c = \bar{y} - m\bar{x}.$$

Check your answer by plotting the line of best fit along with the data.

(b) Given that the data set is a just sample of some underlying distribution, with random noise: What do you think might be the 'true' values of m and c in the underlying model? What is the uncertainty in your best-fit parameters m and c? How could you estimate this uncertainty from the data? Discuss with your neighbour.

---

2. A linear model (i.e. linear in its parameters).

Load the second data set (`ds2.txt`) and plot. We are going to fit the data to a cubic in x:

$$f(x,\beta_j) = \sum_{j=0}^{3} \beta_j \phi_j(x), \qquad \phi_j(x) = x^j.$$

We will solve the **normal equations** for the vector of parameters $\boldsymbol{\beta} = (\beta_0, \ldots \beta_3)$:

$$(\mathbf{X}^T\mathbf{X})\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}.$$

(a) Calculate the $N \times 4$ matrix $\mathbf{X}$ with elements $X_{ij} = \phi_j(x_i)$.

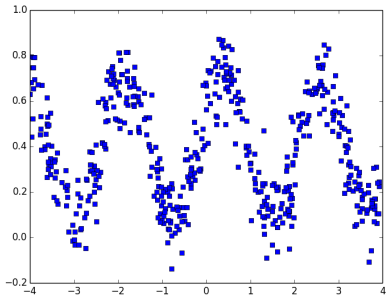
(b) Compute the 4×4 matrix $\mathbf{X}^T\mathbf{X}$ and the vector $\mathbf{X}^T\mathbf{y}$. Hence solve the normal equations for $\boldsymbol{\beta}$ using `np.linalg.solve()`. (You should find $\beta_0 \approx 1.0$, $\beta_1 \approx 1.5$, $\beta_2 \approx 0$ and $\beta_3 \approx -0.2$.)

(c) Fit the same model using the function `scipy.optimize.curve_fit()`. Compare the parameter values obtained with those from part (b) – they should be identical.

```
from scipy.optimize import curve_fit
?curve_fit     # to see the help file / docstring.
```

---

3. A non-linear model. Load `ds3.txt` and plot.



We could try fitting a non-linear model of the following form:

$$f(x, \beta_j) = \beta_0 + \beta_1 \sin(\beta_2 x + \beta_3)$$

Use `curve_fit()` to find best-fit parameters for this model. Check your answers by plotting the model on top of the data.
