MECH3750 PBL Content Summary
Week 2

Content:

- Least Squares

Upcoming assessment:

- Problem Sheet 2 (due before Week 3 PBL session)
- Quiz 1 (Week 3 PBL session)
  - Taylor series
  - Finite differences
  - Newton's method
  - Least squares

Tutors: Nathan Di Vaira, Tristan Samson, Nicholas Maurer, Jakob Ivanhoe, Robert Watt

1 Least Squares Approximation

The least squares method can be used to approximate a vector $\mathbf{f}$ with n vectors $\mathbf{p^{(1)}}, \mathbf{p^{(2)}}, ..., \mathbf{p^{(n)}}$,

$$\mathbf{f} \approx \alpha_1 \mathbf{p^{(1)}} + \alpha_2 \mathbf{p^{(2)}} + ... + \alpha_n \mathbf{p^{(n)}}.$$

We can optimise the approximation by first defining a positive error,

$$E = \sum_i \left(\alpha_1 p_i^{(1)} + \alpha_2 p_i^{(2)} + ... + \alpha_n p_i^{(n)} - f_i \right)^2,$$

for each component of the vector, i.

The error is minimised by taking its derivatives w.r.t. all coefficients α_j ($\forall j \in 1, 2, ..., n$),

$$\frac{\partial E}{\partial \alpha_j} = \sum_i 2 p_i^{(j)} \left(\alpha_1 p_i^{(1)} + \alpha_2 p_i^{(2)} + ... + \alpha_n p_i^{(n)} - f_i \right),$$

and setting to 0,

$$0 = \sum_i p_i^{(j)} \left(\alpha_1 p_i^{(1)} + \alpha_2 p_i^{(2)} + ... + \alpha_n p_i^{(n)} \right) - \sum_i p_i^{(j)} f_i.$$

This can be written in matrix notation using the dot product $\mathbf{f} \cdot \mathbf{g} = \sum_i f_i g_i$,

$$\begin{bmatrix} \mathbf{p^{(1)}} \cdot \mathbf{f} \\ \mathbf{p^{(2)}} \cdot \mathbf{f} \\ \vdots \\ \mathbf{p^{(n)}} \cdot \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{p^{(1)}} \cdot \mathbf{p^{(1)}} & \mathbf{p^{(1)}} \cdot \mathbf{p^{(2)}} & \cdots & \mathbf{p^{(1)}} \cdot \mathbf{p^{(n)}} \\ \mathbf{p^{(2)}} \cdot \mathbf{p^{(1)}} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{p^{(n)}} \cdot \mathbf{p^{(1)}} & \cdots & \cdots & \mathbf{p^{(n)}} \cdot \mathbf{p^{(n)}} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

$$\boldsymbol{\alpha} = (P^T P)^{-1} P^T \mathbf{f}$$

and solved for $\boldsymbol{\alpha}$.

1.1 Regression

Least squares approximation can also be used to fit discrete data points with a continuous function.

For example, to fit n data points (x_i, y_i) with a quadratic function $f(x) = \alpha_1 x^2 + \alpha_2 x + \alpha_3$, the coefficients $\alpha_1, \alpha_2, \alpha_3$ can be found by minimising the error,

$$E(\alpha_1, \alpha_2, \alpha_3) = \sum_{i=1}^{n} \left(\alpha_1 x_i^2 + \alpha_2 x_i + \alpha_3 - y_i\right)^2 .$$

We do this by re-writing in vector notation,

$$E(\alpha_1, \alpha_2, \alpha_3) = \sum_{i=1}^{n} \left(\alpha_3 p_i^{(0)} + \alpha_2 p_i^{(1)} + \alpha_1 p_i^{(2)} - y_i\right)^2 ,$$

where,

$$\begin{aligned}
\boldsymbol{p^{(0)}} &= (1, 1, ..., 1), \\
\boldsymbol{p^{(1)}} &= (x_1, x_2, ..., x_n), \\
\boldsymbol{p^{(2)}} &= (x_1^2, x_2^2, ..., x_n^2),
\end{aligned}$$

and solving $\frac{\partial E}{\partial \alpha_j} = 0$ for $\boldsymbol{\alpha}$, i.e., solving the matrix equations $\boldsymbol{\alpha} = (P^T P)^{-1} P^T \mathbf{f}$ as above, such that,

$$P = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix} .$$

1.2 Fitting Functions

What if we want to approximate a continuous function (e.g. cubic) with another function (e.g. linear) over an interval $[a, b]$?

For an approximating polynomial of degree N, we can define the error,

$$E = \int_a^b \left(\sum_{j=1}^{N} \alpha_j p_j(x) - f(x) \right)^2 dx,$$

where $p_1(x) = 1, p_2(x) = x, ..., p_N(x) = x^{N-1}$.

Again, we solve $\frac{\partial E}{\partial \alpha_j} = 0$ for $\boldsymbol{\alpha}$,

$$\begin{bmatrix} \int_a^b p_1 f dx \\ \int_a^b p_2 f dx \\ \vdots \\ \int_a^b p_N f dx \end{bmatrix} = \begin{bmatrix} \int_a^b p_1 p_1 dx & \int_a^b p_1 p_2 dx & \dots & \int_a^b p_1 p_N dx \\ \int_a^b p_2 p_1 dx & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \int_a^b p_N p_1 dx & \dots & \dots & \int_a^b p_N p_N dx \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}.$$