SOEE3250/SOEE5675 2018/19

Inverse Theory
Andy Hooper

Practical 3: Best Linear Unbiased Estimator (BLUE)

This practical will be assessed. Please submit your answers AND your Matlab code via the VLE by 5 pm, 22nd October. Don't forget to label your figures

The aim is to set up a problem to be solved by the BLUE then demonstrate that the BLUE is unbiased and gives better results than the unweighted least squares estimator. To achieve this, you will simulate measurements for a given set of model parameters and then use the BLUE (as well as unweighted least squares) to invert the problem. Simulation of data has the advantage that you know what the true values are for the model parameters, which enables you to assess how successful the estimator is.

Inverse problem description

The goal is to characterise the subsidence over a gas field that is being exploited. The subsidence is assumed to follow an exponential decay of the form:

$$h = h_0 + A\left(e^{-\frac{t}{150}} - 1\right)$$

where h is height, h_0 is height at time t=0, t is the time in days and A is the total subsidence. A benchmark is installed at the point of maximum subsidence and its height measured using precise GPS every 60 days for 300 days. The errors of the height measurements have zero mean and their standard deviations are:

Time(days)	0	60	120	180	240	300
Standard deviation (cm)	2	4	3	2	4	1

1. Formulate the forward problem in the format **d**=**Gm** (give **m**, and **G**). [4]

2. Assuming h_0=151 m and A=0.18 m, use Matlab to calculate what the measurements would be if there were no measurement errors (**d_true**). This is the forward problem – you are using it to simulate data in this case, which you can then invert. Plot **d_true** against time as blue circles (use the `'o'` option). Remember to label the plot appropriately. [3]

3. What is the variance-covariance matrix for the observations (**Qdd**)? Hint: consider the variances (along the diagonal) and covariances separately. [2]

4. Calculate some random noise to represent the error for each of the measurements, assuming the noise is drawn from a Gaussian (normal) distribution with the appropriate standard deviation according to the table above. To do this, initiate the random number generator using `>>rng(992)`, then generate 6 random noise values using `>>randn` and scale the noise appropriately (initiating the random number generator ensures that whenever the script, is re-run the same sequence of random numbers will be generated). Add the noise to **d_true** to give simulated measurements

(**d**). Add **d** to your plot as green crosses (use >>`hold on` to keep the blue circles). [3]

5. Use the BLUE to estimate the subsidence curve from the simulated measurements. Add the curve to your plot as a red line. [2]

6. Repeat the calculations in questions 4 and 5, 1000 times, with a different realisation of the noise each time (i.e., leave out the >>`rng` step). One way to achieve this is to create a loop with >>`for i=1:1000` and >>`end.` Save the estimated model parameters for each iteration, but do not add points/lines to your plot.

   Plot histograms (use >>`hist` with appropriate number of bins) for each of the model parameters. Add the true values for the model parameters to the histograms as vertical red lines. Is the estimator unbiased? (explain your answer briefly) [4]

7. Plot the 1000 values of the 2 model parameters against each other as a scatter plot (use >>plot with the `'.'` option). Are they correlated? (explain your answer briefly) [2]

8. Calculate the variance-covariance matrix for the model parameters (**Qmm**) using propagation of errors. Also, calculate the sample variances of each model parameter from the 1000 iterations. Are they consistent with **Qmm** (explain briefly)? [4]

9. Repeat the 1000 iterations, but use unweighted least squares as the estimator instead of the BLUE. Calculate the sample variances for each model parameter. Are they better or worse than the variances for the BLUE results, and is this as you expect? [4]