Propagation of Errors in Least Squares

There are three questions to consider when fitting parameters. These are

1. What is the best choice of parameters?
2. What are the errors estimates for the parameters?
3. What is the statistical measure of the goodness-of-fit?

We have already discussed how to use least squares to find the choice of parameters that maximize the likelihood of the data. We now address the second and third questions.

Normally distributed random variables are stable in the sense that the sum of two independent normally distributed random variables is another normally distributed random variable. Moreover, if w_i are weights and Y_i are independent normally distributed random variables with with mean f_i and variance σ_i^2, then $Z = \sum_{i=1}^n w_i Y_i$ is a normally distributed random variable with mean $\sum_{i=1}^n w_i f_i$ and variance $\sum_{i=1}^n w_i^2 \sigma_i^2$. We shall use this fact to calculate how independent normally distributed measurement errors propagate to error estimates for the solution of the least squares problem.

Recall that the best choice of parameters c is given by the maximum likelihood estimator $c = R_1^{-1} Q_1^T \tilde{y}$ where R_1 is an $m \times m$ upper triangular matrix, Q_1 is an $n \times m$ matrix with orthogonal columns and the data points $\tilde{y}_i = y_i/\sigma_i$ have be rescaled to be independently normally distributed with variance 1. Since each c_j is a weighted sum of the data points $\tilde{y}_i$ then each c_j also has a normally distributed error. Thus, we may calculate the variance $\sigma(c_j)^2$ of c_j as

$$\begin{aligned}
\sigma(c_j)^2 &= \sum_{i=1}^n [R_1^{-1} Q_1^T]_{ji}^2 = \sum_{i=1}^n \Big(\sum_{k=1}^m [R_1^{-1}]_{jk} [Q_1]_{ki} \Big)^2 \\
&= \sum_{i=1}^n \sum_{k=1}^m \sum_{l=1}^m [R_1^{-1}]_{jk} [Q_1^T]_{ki} [R_1^{-1}]_{jl} [Q_1^T]_{li} \\
&= \sum_{k=1}^m \sum_{l=1}^m \Big(\sum_{i=1}^n [Q_1^T]_{ki} [Q_1]_{il} \Big) [R_1^{-1}]_{jk} [R_1^{-1}]_{jl} \\
&= \sum_{k=1}^m \sum_{l=1}^m [I]_{kl} [R_1^{-1}]_{jk} [R_1^{-1}]_{jl} = \sum_{k=1}^m [R_1^{-1}]_{jk}^2 = [R_1^{-1} (R_1^{-1})^T]_{jj}
\end{aligned}$$

In statistics, the matrix $V = R_1^{-1}(R_1^{-1})^T$ is called the covariance matrix. The diagonal entries of V represent the variances of c_j and the off-diagonal entries the covariances.

Continuing Matlab Example 19a, we can form the covariance matrix by

Matlab Example 20a

```
>> [Q1,R1]=qr(Atilde,0);
>> R1inv=inv(R1);
>> V=R1inv*R1inv'
V =
   1.3971647  -0.1825548  -0.2272652   0.0422962
  -0.1825548   0.3979981   0.0686120  -0.0329736
  -0.2272652   0.0686120   0.0652944  -0.0146202
   0.0422962  -0.0329736  -0.0146202   0.0045669
```

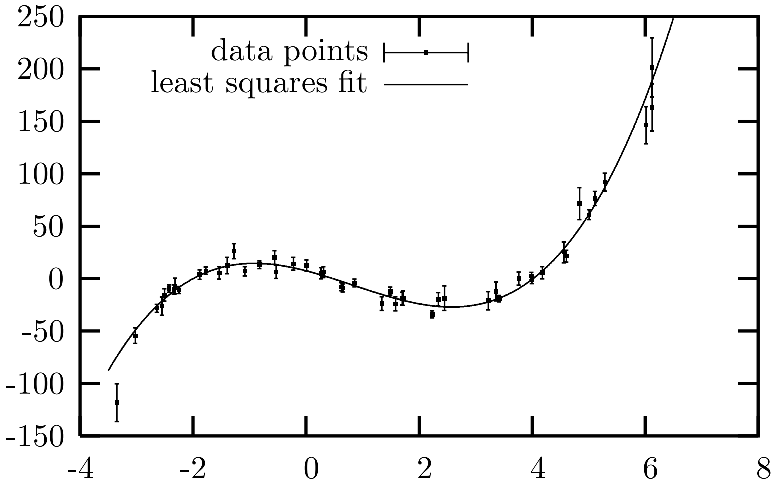
The diagonal entries of this matrix are the variances $\sigma(c_j)^2$ of the c_j. An estimate of the errors in the c_j's is then given by the standard deviations $\sigma(c_j)$ found by taking the square roots of the diagonal entries of V. This is shown in

Matlab Example 20b

```
>> stddev=sqrt(diag(V))'
stddev =
  1.182017  0.630871  0.255528  0.067579
```

We end with a discussion of a statistical measure of the goodness-of-fit. First let us examine the graph of the data points and the least squares approximation we just found.

Graph of Least Squares Fit



It's tempting to say, it looks good, and be done. However, statistics can tell us more.

Let n be the number of data points and m be the number of parameters. The probability that the minimum of χ^2 is less than C^2 is given by

$$P = P\{\,\chi^2 < C^2\,\} = \frac{1}{\Gamma(a)} \int_0^{C^2/2} e^{-t} t^{a-1} dt$$

where $a = (n-m)/2$. Therefore, if we set $C^2 = \|\tilde{A}c - \tilde{y}\|_2^2$ for the optimal parameter c found for the observed data points y_i, then $Q = 1 - P$ should be substantially greater than 0. As a rule of thumb, Q much less than 10^{-3} is considered a bad fit. This provides a quantitative measure of the goodness of fit.

The integral defining P is known as the incomplete gamma function. We will learn numerical quadrature rules for computing such integrals later in this course. For now, note that just like `log` and `sin`, Matlab has this particular function, called `gammainc`, built in. Thus, to compute Q we may write

Matlab Example 20c

```
>> C2=norm(Atilde*c-ytilde)^2
C2 = 54.361
>> Q=1-gammainc(C2/2,(length(ytilde)-length(c))/2)
Q = 0.18608
```

Since this value of Q is far from zero, then we conclude that the data fits the model.