Confidence intervals and more Ch. 6 & 7

Bob Douma & Alejandro Morales

28 November 2017

Optimisation problems and assessing the confidence limits of parameter estimates

Fitting a model to data requires you to specify a relationship between variables. After specifying this relationship we need to fit parameters of this model that best fits the data. This fitting is done through computer algorithms (optimizers). However, sometimes it may be hard to fit a model to data. After having found the best fitting model, you want to assess how certain you are about the parameter estimates. For assessing the uncertainty of model parameters several methods exist that have pros and cons.

This exercise will have two purposes. First you will learn that an innocent looking function can be challenging to fit. Second, you will learn to assess the uncertainty in the parameter values. For assessing the uncertainty in the parameter estimates there are two methods: the profiling method and the quadratic approximation. Bolker recommends to use the likelihood profile for assessing the uncertainty in the parameters because this one is more accurate than the one based on the Hessian matrix.

1. Take the first dataset of the six datasets you have worked with earlier on. Assume that the function was generated by the monomolecular function $a(1 - e^{(-bx)}$. Fit this model with normally distributed errors through this data with `mle2` and optim method `Nelder-Mead`. Choose four different starting points of the optimisation: `start_a = c(5,10,20,30)`, `start_b = c(0.001,0.005,0.01,0.1)` and compare the NLL of those four optimisations. Plot the curves into the plot with data and try to understand what happened.

2. To understand the behaviour of the optimisation routine we will plot the likelihood surface over a range of values of a and b. For a choose a number of parameter values in the range of 0-40 and for b choose a number of values in the range 0.1-10. Calculate for each combination the NLL and plot the NLL surface using `contour` plot. For more insight into the functioning of what the optimisation method did, you can add the starting points that you gave to mle2 and the best fitting points, use `points()` for this. Do you have a clue why the optimisation did not find the minimum point in the landscape? Now zoom in and choose values for b in the range of 0.001-0.03 and check again the NLL surface.

   hint: See Bolker Lab 6 for inspiration on coding.

   hint: You can use a for a double for-loop to run over all parameters

   hint: Store the NLL results in a matrix (you can make a 100x100 matrix by `matrix(NA,nrow=100,ncol=100)`).

3. Calculate the confidence intervals of the parameters through constructing the likelihood profile. Consult page 106 of Bokler or Lab 6 for how to calculate the confidence intervals based on the likelihood profile. Use the following pseudocode to achieve this:

   a. Adapt the likelihood function such that one parameter is not optimised but chosen by you, say parameter a.
   b. Vary a of a range and optimise the other parameteters.
   c. Plot the NLL as a function of parameter a.
   d. Find the values of a that enclose $-L + \chi^2(1 - \alpha)/2$. In `R` this can be done through `qchisq(0.95,1)/2`.
   e. Compare your results with the results from the `R` function `confint()`. `confint()` uses the profiling method along with interpolation methods.

4. *(time permitting)* Calculate the confidence intervals through the quadratic approximation. Take the following steps to achieve this:

a. Get the standard error of the parameter estimates through `vcov`. Note that `vcov` return the variance/covariance matrix
b. Calculate the interval based on the fact that the 95% limits are 1.96 (qnorm(0.975,0,1)) standard deviation units away from the mean.

5. *(time permitting)* Plot the confidence limits of the both method and compare the results. Is there a big difference between the methods?

6. To assess the uncertainty in the predictions from the model you can construct population prediction intervals (PPIs, see 7.5.3 Bolker). Population prediction intervals shows the interval in which a new observation will likely fall. To construct the PPI take the following steps

   a. Simulate a number of parameter values taken the uncertainty in the parameter estimates into account.

      hint: If the fitted mle object is called `mle2.obj`, then you can extract the variance-covariance matrix by using `vcov(mle2.obj)`. You can extract the mean parameter estimates by using `coef(mle2.obj)`. Now you are ready to simulate 1000 combinations of parameter values through `z = mvrnorm(1000,mu=coef(mle2.obj),Sigma=vcov(mle2.obj))`. `mvrnorm` is a function to randomly draw values from a multivariate normal distribution.

   b. Predict the mean response based on the simulated parameter values and the values of x

      hint: make a for-loop and predict for each simulated pair of parameter values the mean for a given x. Thus `mu = z[i,1]*(1-exp(-z[i,2]*x))`

   c. Draw from a normal distribution with a mean that was predicted in the previous step and the sd that you simulated in step a.

      hint: `pred = rnorm(length(mu),mean=mu,sd=z[i,3])`. Store pred in a matrix with each simulated dataset in a seperate row.

   d. Calculate for each value of x the 2.5% and the 97.5% quantiles

      hint: If the predictions are stored in a matrix `mat`, you can use `apply(mat,2,quantile,0.975)` to get the upper limit.

Hints for choosing deterministic functions and stochastic functions

1. Deterministic functions

   dataset 1 light response curve. There are a number of options of functions to choose from, depending on the level of sophistication: $\frac{ax}{(b+x)}$, $a(1-e^{(-bx)})$, $\frac{1}{2\theta}(\alpha I + p_{max} - \sqrt{(}\alpha I + p_{max})^2 - 4\theta I p_{max})$ see page 98. A parameter `d` can be added in all cases to shift the curve up or down.

   dataset 2 The dataset describes a functional responses. Bolker mentions four of those $min(ax, s)$ $\frac{ax}{(b+x)}$, $\frac{ax^2}{(b^2+x^2)}$, $\frac{ax^2}{(b+cx+x^2)}$

   dataset 3 Allometric relationships generally have the form ax^b

   dataset 4 This could be logistic growth $n(t) = \frac{K}{1+(\frac{K}{n_0})e^{-rt}}$ or the gompertz function $f(x) = e^{-ae^{-bx}}$

   dataset 5 What about a negative exponential? $ae{-}bx$ or a power function ax^b

   dataset 6 Species reponse curves are curves that describe the probability of presence as a function of some factor. A good candidate good be a unimodel response curve. You could take the equation of the normal distribution without the scaling constant: e.g. $ae^{\frac{-(x-\mu)^2}{2\sigma^2}}$

2. Stochastic functions/Probability distributions

   dataset 1 y represents real numbers and both positive and negative numbers occur. This implies that we should choose a continuous probability distribution. In addition, the numbers seems unbound. Within the family of continuous probability distributions, the normal seems a good candidate distribution because this one runs from -inf to +inf. In contrast the Gamma and the Lognormal only can take positive numbers, so these distributions cannot handle the negative numbers. In addition, the beta distribution is not a good candidate because it runs from 0-1.

   dataset 2 y represents real numbers and only positive numbers occur. The data represents a functional response (intake rate of the predator), and it is likely that you can only measure positive numbers (number of prey items per unit of time). This implies that we should choose a continuous probability distribution. Within the family of continuous probability distributions, the Gamma and the Lognormal could be taken as candidate distributions because they can only take positive numbers (beware that the Gamma cannot take 0). However, you could try to use a normal as well.

   dataset 3 y seems represents counts (this is the cone dataset that is introduced in ch. 6.). Given that it contains counts we can pick a distribution from the family of discrete distributions. The Poisson and the Negative Binomial could be good candidates to describe this type of data.

   dataset 4 y represents population size over time. From looking at the data, they seems to represent counts. Given that it contains counts we can pick a distribution from the family of discrete distributions. The Poisson and the Negative Binomial could be good candidates to describe this type of data.

   dataset 5 No information is given on y. The data clearly seems to represent counts. Thus the same reasoning applies here as to the two previous datasets.

   dataset 6 The data (y) represents species occurences (presence/absence). The binomial model would be a good model to predict the probability of presence.