Maximum-Likelihood Fitting

Nathaniel Phillips

8 Sep 2015

The purpose of this document is to demonstrate the steps in calculating Maximum Likelihood (ML) estimates of parameters of models given data. I'll start by showing how to do this with the simple linear model. I'll then show how to calculate ML parameter estimates for Gonzalez-Vallejo's (2001) Proportional Difference (PD) model.

Three basic steps

There are three basics steps to obtaining ML estimates for model parameters

1. Get (or simulate) **data**

Pretty straighforward. If you don't have data yet, you can simulate data and see how well you can recover model parameters.

2. Define the **Loss Function**

Defining the loss function is the most important part of the ML process. A loss function takes model parameters and data as input, and returns a quantitative value (a loss value) that indicates how consistent the data are with the model given specified parameter values. With ML estimation, we will define the loss function as the *deviance*, -2 times the sum of the log-likelihoods of the data. Formally:

$$dev = -2 \times \sum_{i=1}^{N} ln(l_{\vec{p}}(x_i))$$

Where $l_{\bar{p}}(x_i)$ is the likelihood of the ith data point given a vector of parameter values $\vec{p}$, and *N* is the total number of data points. Likelihoods are defined by specific probability distributions, such as the Binomial for binary (0 or 1) data or Normal for continuous data. The specific model you are testing will specify the exact likelihood for any given data point.

3. **Optimize**: Find parameter values that minimize the loss function.

Once you have data and have defined the loss function, you can use optimization procedures to find the specific parameter values that minimize the loss function (and simultaneously maximize the log-likelihood of the data). There is no *one optimization procedure to rule them all*. No optimization procedure is perfect - that is, none can guarantee to find the best parameter values. However, for this tutorial we'll use the default optimization procedures built in to the *optim* function in R.

Example 1: Linear regression with 1 predictor

To start, let's calculate ML estimates for the simple linear model. I'll generate random predictors x, then noisy responses y. We will assume the underlying model is y ~ ax + b + e. Our goal is to calculate ML estimates for a, b and the standard deviation of errors (e)

First, let's create some data. We'll have data.x be the independent variable, and data.y be the dependent variable.

I'll set data.x to 100 random Normal samples with mean 10 and sd 2.

```
data.x <- rnorm(n = 100, mean = 10, sd = 2)
```