
School of Informatics, University of Edinburgh
Frank Keller

Solutions for Tutorial 2: Maximum Likelihood Estimation

Please work through this tutorial sheet on your own time as much as possible before arriving in tutorial. We encourage you to work together and discuss your methods and solutions.

1 Maximum Likelihood

Imagine you have conducted an experiment on memory recall. You have developed a novel model that you believe can explain the behavior you found in the experiment. Instances of your model are specified by a set of parameters.

Question 1: Within this context, what is the goal of maximum likelihood estimation (MLE)?

Solution 1: The goal of MLE is to estimate optimal values for each of the model’s parameters by maximizing the likelihood of the data given the parameters. MLE provides an answer to the question “which model instance best fits the data?” when “best fits” is defined as “gives the most evidence for”. If $\theta$ is a particular assignment of values to the model’s parameters and $y$ represents the observed data then the MLE answer to this question is given by

$$\hat{\theta} = \arg\max_{\theta} L(\theta|y) = \arg\max_{\theta} P(y|\theta)$$

Question 2: Briefly describe an alternative way of fitting model parameters and compare it to MLE. Why might you choose one over the other?

Solution 2: Estimating optimal parameters can also be achieved by minimizing the error of a model’s predictions by minimizing the Root Mean Squared Deviation (RMSD) between the observed data and the model’s predictions. For example, predicting how often people can recall 10 items in a particular sequence and then comparing those predictions to how well people actually complete this task. If $y$ are the actual observed data, $M(\theta)$ are the predictions of the model with parameters $\theta$, and $N$ is the number of observations then we want to find

$$\hat{\theta} = \arg\min_{\theta} \text{RMSD}(\theta) = \arg\min_{\theta} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - M(\theta)_i)^2}$$

We may prefer to use MLE over error minimization because MLE is rooted in statistical theory and comes with some attractive properties such as parametrization invariance, consistency, efficiency, and asymptotic normality. RMSD minimization may be preferable to those familiar with linear regression.

Question 3: Could you compute the probability of a particular set of parameter values for your model, given the data, using only its likelihood function? If so, how? If not, why not?
Solution 3: Likelihoods are not sufficient to compute the probability of a particular set of parameter values given the data. By Bayes rule, the probability of the parameters given the data is

\[ P(\theta|\mathbf{y}) = \frac{P(\mathbf{y}|\theta)P(\theta)}{P(\mathbf{y})} \]

but the likelihood of the data given the parameters

\[ L(\theta|\mathbf{y}) = P(\mathbf{y}|\theta) \neq P(\theta|\mathbf{y}) \]

and MLE is of no help because it identifies the set of optimal parameter values but not a probability for that particular set.

We would need to know/specify a prior probability of the parameters \( P(\theta) \) and compute the “evidence” for the data \( P(\mathbf{y}) = \int_\mathcal{A} P(\mathbf{y}|\theta)P(\theta)d\theta \) in order to compute the conditional probability of some set of parameters given the data.

2 MLE on the Ex-Gaussian Function

Download the following file, which contains the model that you will work with in this tutorial (it is linked from the Assignments page):

http://www.inf.ed.ac.uk/teaching/courses/ccs/tutorials/ccs_t02.tar.gz

Unpack this file, which will result in three Matlab source files:

- `exGaussPDF.m` implements the ex-Gaussian function introduced in the lecture;
- `Lopt.m` defines the joint likelihood function for the ex-Gaussian function and uses the Nelder-Mead algorithm to find its maximum;
- `LSurfaceL.m` plots the likelihood surface and runs the maximization.

The code is taken from Lewandowsky and Farrell (2011, ch. 4) and explained there in more detail. Please read this chapter if you need help understanding the code.

Now run `LSurfaceL.m`, which will generate a plot of the likelihood surface. You can examine it using the “Rotate 3D” button in the Matlab GUI.

**Question 4:** What is the shape of the likelihood surface? Try to estimate the maximum by eyeballing the plot.

**Solution 4:** The maximum is approximately \((\mu, \tau) = [3 \, 2.5]\).

Running `LSurfaceL.m` also runs the Nelder-Mead optimization algorithm and outputs both the operations performed and the optimum the algorithm finds.

**Question 5:** Examine the sequence of operations. Does the algorithm generate plausible output? Does it converge on a minimum for the likelihood function? Is the minimum the same as the one you estimated by eyeballing the plot?

**Solution 5:** Yes, the algorithm generates a plausible sequence of reflect, expand, contract operations. The minimum it finds is \((\mu, \tau) = [2.9412 \, 2.4755]\), which is close to what was obtained by eyeballing the plot.
Look at the code of LSurfaceL.m. In line 30, it calls the function Lopt which runs the maximum likelihood optimization. The start values for the parameters are \((\mu, \tau) = [0.1 \ 0.1]\).

**Question 6:** Vary the start values of the parameters. Does the algorithm always converge on the same answer? Which results do you get for \([0 \ 1], [1 \ 0], [1 \ 2], [0 \ 0]\)?

**Solution 6:** The algorithm gives the same results for \([0 \ 1]\) and \([1 \ 2]\), but the likelihood is 0 for \([1 \ 0]\) and \([0 \ 0]\) (probably due to numerical underflow), so Nelder-Mead can’t operate.

### 3 The Role of the Data for MLE

We will now investigate how the likelihood function \(L(\theta | y)\) is influenced by the data \(y\) used to compute it. The code for LSurfaceL.m specifies \(rt\), the vector of data points on which the likelihood estimation is carried out.

**Question 7:** What happens if the data vector \(rt\) is changed to contain a single point? Try \(rt = [0.1], rt = [1], rt = [10]\). Explain the change in the likelihood surface. What happens for \(rt = [100]\)?

**Solution 7:** The result is a very peaked likelihood function, with a maximum whose value of \(\mu\) matches the value of the data point closely, and with \(\tau\) close to 0. Note that for high data values \(rt = [10], rt = [100]\), the visualization is out of range. To fix this, adjust the value of \(mMu\) in LSurface.m.

As explained in the lecture, in the general case, we will have multiple data points, and we are optimizing the joint likelihood of all the data points.

**Question 8:** Plot the likelihood surfaces for the following data vectors and compare them the ones you obtain if you have only a single data points (as in the previous question).

\[
rt = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\
rt = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 10] \\
rt = [1 \ 1 \ 1 \ 1 \ 1 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10] \\
rt = [3 \ 4 \ 5 \ 6 \ 6 \ 7 \ 8 \ 8 \ 9 \ 10 \ 11]
\]

**Solution 8:** The first data vector results in an even more steeply peaked likelihood surface, as there is only one values of \(\mu\) that matches the data well, even nearby values lead to a big reduction in likelihood. This is because we are visualizing the joint likelihood \(L(\theta | y) = \prod_{i} L(\theta | y_i)\). This quantity will be larger for the optimal \(\theta\) when the product ranges over multiple identical data points compared to when it ranges over multiple different data points.

The second data vector essentially contains an outlier, which makes the likelihood surface more spread out.

The third data vector requires a different model to be fit well, for example a mixture of two Gaussians. For the present model, it will just result in a fairly flat likelihood surface, as there are a number of parameter settings that are compatible with this distribution of the data.

The last data vector looks unproblematic (and the likelihood surface doesn’t seem unusual), but the optimization algorithm doesn’t converge; it return a likelihood of 0. It’s not clear why; this might be a limitation in the Nelder-Mead algorithm.
In the current implementation, the maximum likelihood estimation for the ex-Gaussian function only adjusts two parameters, $\mu$ and $\tau$. Extend the code for Lopt.m and LSurfaceL.m so that the third parameter $\sigma$ is also optimized.

**Question 9:** Run the new version of the code on the data vectors from the previous questions. Do you obtain a better fit using three parameters? How can you tell if your fit has improved?

**Solution 9:** Yes, the three-parameter version of the model fits the data better. The likelihood increases from $-8.29617 \cdot 10^{-11}$ to $-1.38069 \cdot 10^{-10}$. However, the AIC and BIC do not improve (become smaller) due to the additional parameter, see lecture 7 (which uses this example).

**Question 10:** Can you think of a way of plotting a likelihood surface that depends on three parameters?

**Solution 10:** You could plot the parameters pairwise (holding the third parameter constant). Or you could use color to represent the fourth dimension (hotter colors correspond to higher values of $\sigma$, etc.). Or you could try to draw multiple surfaces (e.g., 10 graphs for 10 values of $\sigma$), either in the same graph or in multiple graphs.

**References**