# Ps 7 Problem 1 and 2

Adam Guerin

November 14, 2023

**Abstract**

This short paper will show some plots and discuss implementing a minimizing function that uses Brent's method and comparing it to scipy's library equivalent as well as discussing a likelihood maximization problem for a sample data set.

## 1 Introduction

Optimization and minimization are not trivial problems. Entire books and fields of study are dedicated to various algorithms and methods towards finding the minimum value in some situation. There are ways to guarantee finding the minimums for certain subclasses of functions. In these cases it makes sense to combine methods that are fast and unreliable and methods that are reliable and slow into one algorithim, in this case, Brent's method.

Minimization is also used to fit data, in the second problem I do this for a sample dataset and fit it to a logistic function as is typical for yes no questions.

## 2 Methods

Brent's method works by combining golden mean search, which guarantees finding the extremum in some interval, and a parabolic search that fits a parabola to your search interval (for one dimensional functions). The parabolic search is much faster but also much more unreliable and does not guarantee finding the extremum you want. Smart implementation of Brent's combines these and let's me find the minimum of the function $f(x) = (x - .3)^2 e^x$ quickly and easily (after implementing). Brent's method decides on using parabolic or golden mean search by choosing parabolic if it's well behaved (step moves you towards extremum), but otherwise doing golden (which is guaranteed to mvoe you towards the extremum).

For the second problem I optimized a data set of yeses and nos to logistic function using scipy optimize which is better than anything I could write. The logistic function goes as follows:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{1}$$

Where p(x) is the probability of a yes for a given input x. In our case it is the probability of someone of age x recognizing the phrase "Be King, Rewind." As this has to do with VCRs and renting physical movies the older someone is the more likely they would be to recognize it.

## 3 Results

My Brent's method application works, but not quite as well as the scipy implementation, both give roughly the same minimum value. My method returning 0.3000000527406742 in 15 steps and the scipy method returning 0.3000000120362894 in 14 steps too. This is for the same tolerance. The difference between the two and number steps may be due to how I define an iteration or when checks are made, but they agree well. My method was also much more unstable, and would occassionally give wild answers or converge poorly.

The fitting makes sense with the most cross over from half understanding to half not understanding occuring at $\frac{-\beta_0}{\beta_1}$ which is 51 years old. My $\beta_0$ is about -5.62 with an error of about .0202 and my $\beta_1$ is about 0.1096 with an error of about 0.00376. The covariance matrix between these is $\begin{bmatrix} 4.2e-04 & 1.83e-05 \\ 1.83e-05 & 1.41e-05 \end{bmatrix}$ This is supposed to be symmetric, but I can't figure out where I went wrong.

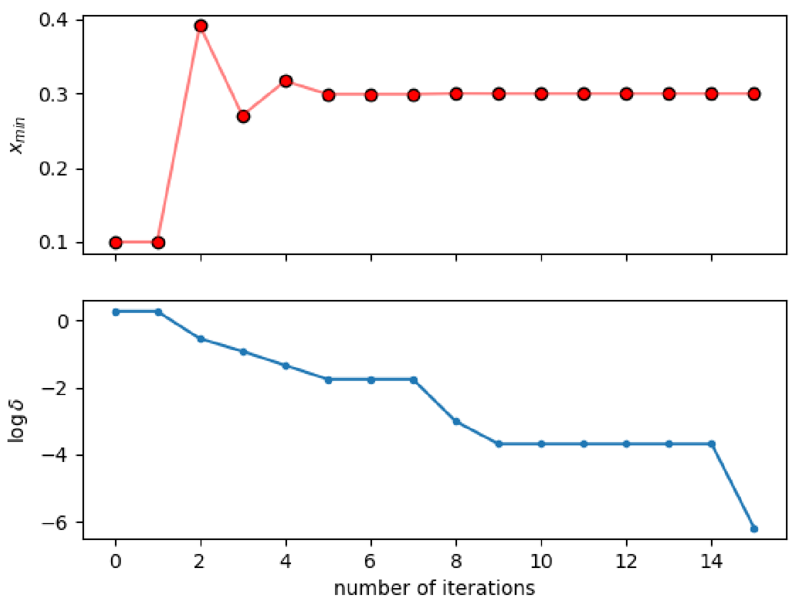


Figure 1: From this we can see the the graph quickly jumping to near the minimum then spending the next grouping of iterations to get within our set tolerance.

## 4 Discussion

Again, minimization and optimization are non trivial, but very useful when you can get it to work. Hopefully your problem isn't so difficult that you can get a paper out of it after finishing it like a lot of the high dimension machine learning and other people are doing.

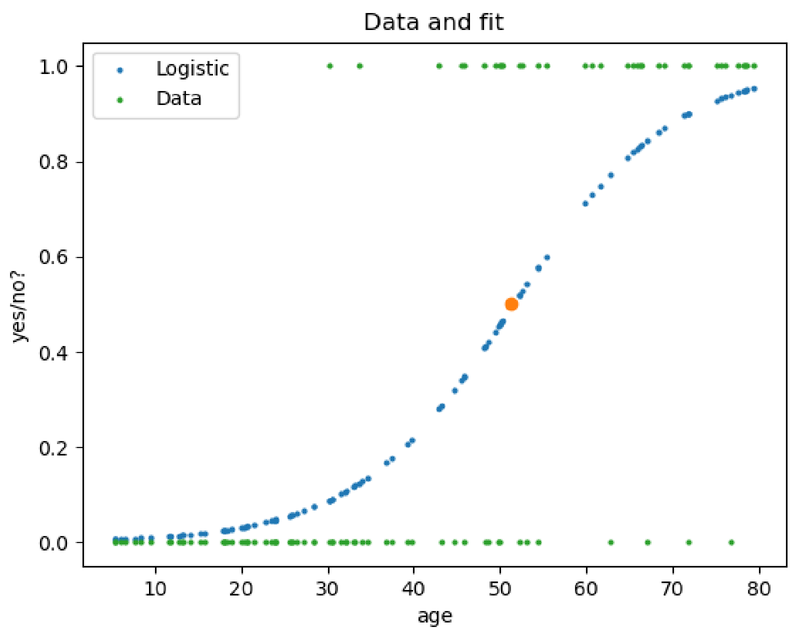


Figure 2: Data and fit plotted onto same plot. This logistic fit makes sense to me, you can see the switch over from mostly no to mostly yes at the point where the data becomes more dense at at yes than at no.