Problem Set 7

Debendro Mookerjee
GitHub Account Name: Debendro-dm5790

November 10, 2023

Abstract

In this problem set we explore optimization in Python by implementing and applying Brent's algorithm to determine the local maximum and minimum of the function $f(x) = (x - 0.3)^2 \exp x$ and by performing the minimization of the negative log likelihood function for logistic regression. We learn about various techniques that aid computational optimization such as plotting the function to be optimized in order to get an idea of where the optima are. We use this when we plot the function $f(x) = (x - 0.3)^2 \exp x$ to determine where to bracket and when we make plots of the log likelihood function to determine roughly where it is maximal.

1 Introduction

Optimization of functions is not just used in determining the local extrema of one-dimensional functions, which we perform in the first exercise of this problem set. A practical application of functional optimization is maximum likelihood estimation (MLE), which is the focus of the second exercise. Given data and a particular type of model we want to fit the data to, MLE is used to determine the best parameter values of that model. This goal is achieved by maximizing a likelihood function which guarantees that for a chosen type of statistical model the observed data is most probable. For this problem set, the chosen modeled type is the logistic function, whose defining equation is as follows:

$$p(x) = \frac{1}{1 + \exp[-(\beta_0 + \beta_1 x)]}$$

The model parameters whose optimal values are sought are β_0 and β_1. The data is bivariate in nature, where the independent variable (x) is age of survey participants in years and the dependent variable is their answer, a yes (1) or a no (0). The likelihood, according to this model, for a yes (no) at a particular age is p(age) (1 - p(age)).

If we represent the response data as $\{y_i\}_{i=1}^n$ and the age data as $\{x_i\}_{i=1}^n$, it follows immediately that the likelihood function is

$$L = \prod_{i=1}^{n} \left[p(x_i)^{y_i} \times (1 - p(x_i))^{1-y_i} \right]$$

The log likelihood function is the following:

$$\ln(L) = \sum_{i=1}^{n} \left[y_i \ln p(x_i) + (1 - y_i) \ln(1 - p(x_i)) \right] = \sum_{i=1}^{n} \left[\ln(1 - p(x_i)) + y_i \ln \left[\frac{p(x_i)}{1 - p(x_i)} \right] \right]$$

This is the function to be maximized. Maximizing this is the same as minimizing the negative log likelihood function.

In this report we describe our methods as well as give our outputs.

2 Methods

2.1 Exercise 1:

In order to implement Brent's optimization algorithm, we first implement a function that defines the equation $f(x) = (x - 0.3)^2 \exp(x)$. We then implement a function that carries out quadratic bracketing, which works as follows. Given three points that make up a quadratic bracket, we determine the corresponding function values at these three bracket points. We then fit a quadratic model that fits these three pairs and determine the extremum x-value for that quadratic model. This extremum is also called a step value. The function determines whether this step value falls outside of the bracket or is greater than the previous previous step. If one of these is true, the function updates the value of a variable called status to let the computer program know that quadratic optimization and bracketing needs to be stopped. This function returns the step value and status.

After implementing the quadratic bracketing function, we implement golden section optimization. We also plot the function in order to determine approximatly where the extrema are and to determine the nature of these extrema. The plot reveals the function has a local minimum and a local maximum. The appropriate initial brackets are chosen and in separate files we determine the minimum and maximum with our implementation of the Brent algorithm. We also compare our results with Python's implementation of the Brent algorithm. We find that there is great agreement.

2.2 Exercise 2:

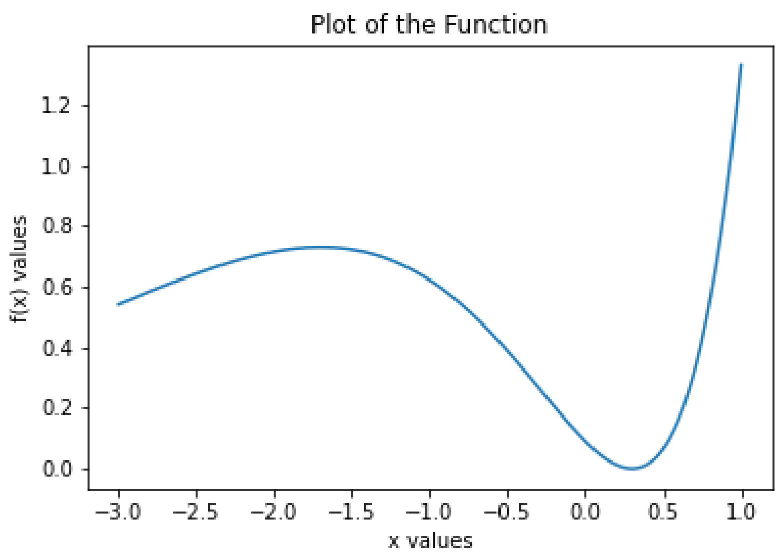
For the MLE problem applied to a logistic model, we first implement the logistic model as a python function and then retrieve the survey information from the appropriate data file. We plot the survey results. We also play around with the logistic model and MLE by plotting the original data, the logistic model as a function of the two parameters, the log likelihood as a function of the two parameters, the two gradients of the log likelihood function, and the four components of its hessian. All of these plots, except for the first one, are plotted as density plots.

In a separate file, we minimize the negative log likelihood function and determine the optimal parameters and the corresponding errors, which are determined by the diagonal elements of the covariant matrix. We also determine the covariant matrix and plot the optimal model and the survey data in one graph. We also determine the value of the negative log likelihood function at these optimal parameter values.

3 Results

3.1 Exercise 1:

Here is a plot of the function, which is helpful in determining where the initial brackets should be.



We see the minimum is located around $x = 0.3$ and the maximum is located around $x = -1.7$

Here is the Brent algorithm for the minimum.

```
Still Using Parabolic Approximtations
Still Using Parabolic Approximtations
Using parabolic steps is not appropriate. Going to perform golden section search.
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Minimum based on my implementation of Brent is at 0.2999999839153427 and function value is 3.4923033505146104e-16
Minimum based on Python implementation of Brent is at 0.3000000005013143 and function value is 3.392411848718073e-19
```

We see when our algorithm goes from quadratic bracketing to golden section search. We also see that there is good agreement between the minimum's location and value we obtained and the ones obtained from Python's Brent's algorithm.

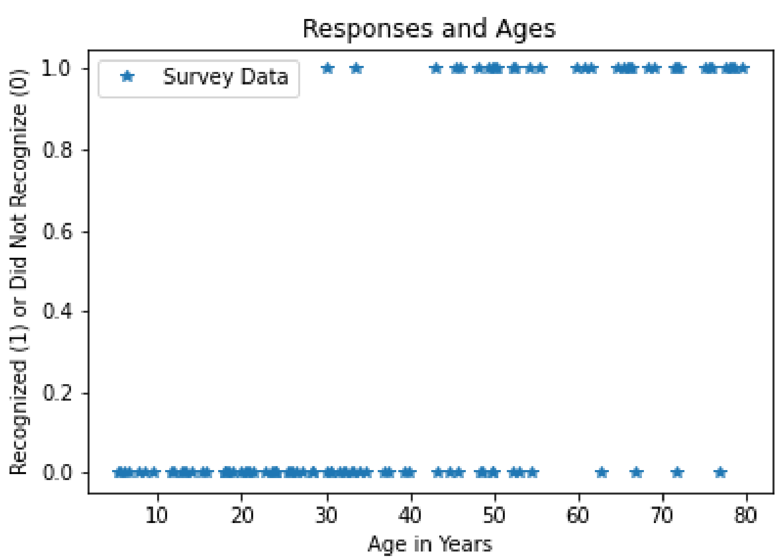
Here is the Brent algorithm for the maximum.

```
Still Using Parabolic Approximtations
Using parabolic steps is not appropriate. Going to perform golden section search.
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Using Golden Section search
Maximum based on my implementation of Brent is at -1.6999992239657091 and function value is
0.7307340962108286
Maximum based on Python implementation of Brent is at -1.6999999946686621 and function value is
0.7307340962109387
```

We see when our algorithm goes from quadratic bracketing to golden section search. We also see that there is good agreement between the maximum's location and value we obtained and the ones obtained from Python's Brent's algorithm.

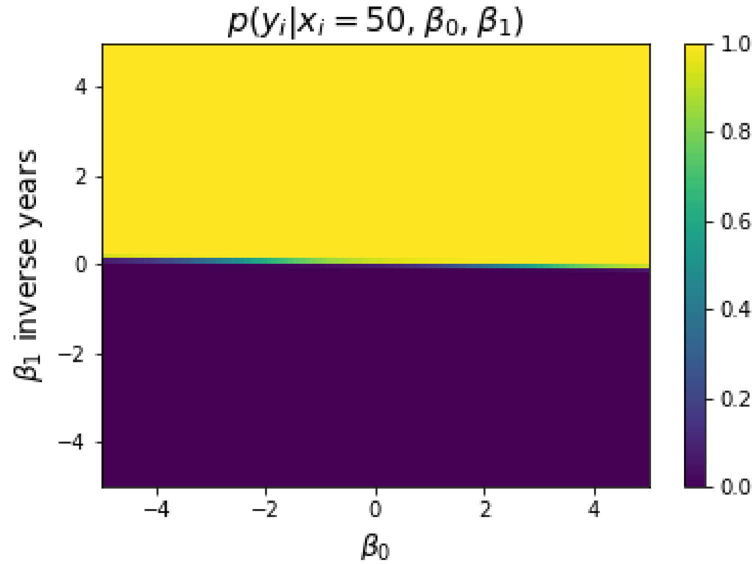
3.2 Exercise 2:

Here is a plot of the survey data. This allows us to determine whether a logistic model is appropriate.



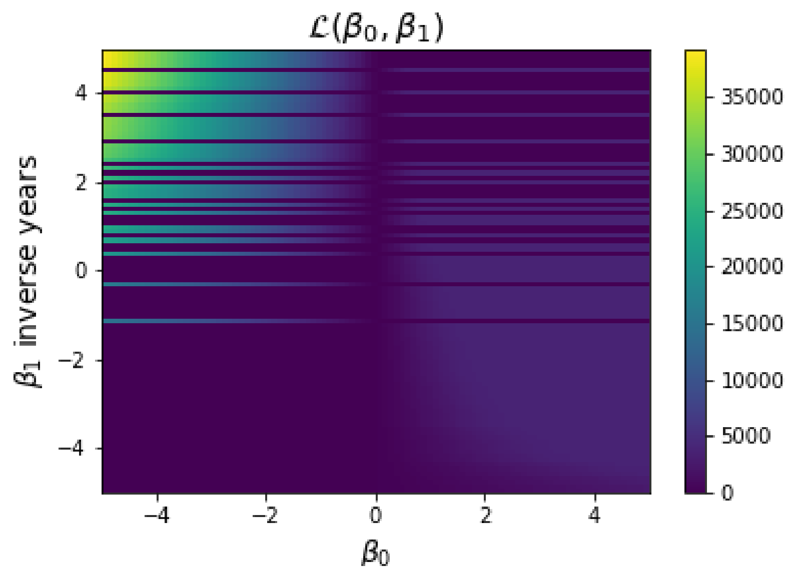
Survey Data. We see the bivariate data as a logistic-like behavior and so MLE with a logistic model is not inappropriate.

Here is a plot of the logistic model as a function of its two parameters β_0 and β_1 when x is fixed at 50.



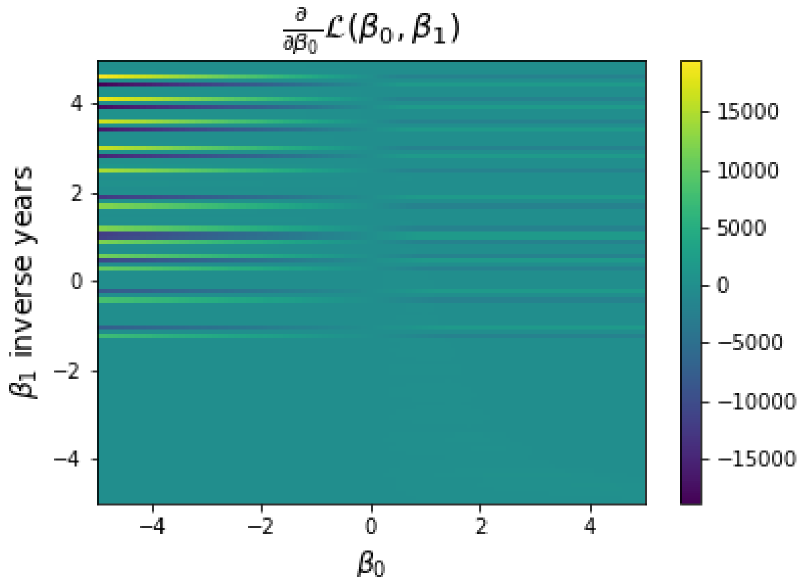
Logistic Model Density Plot

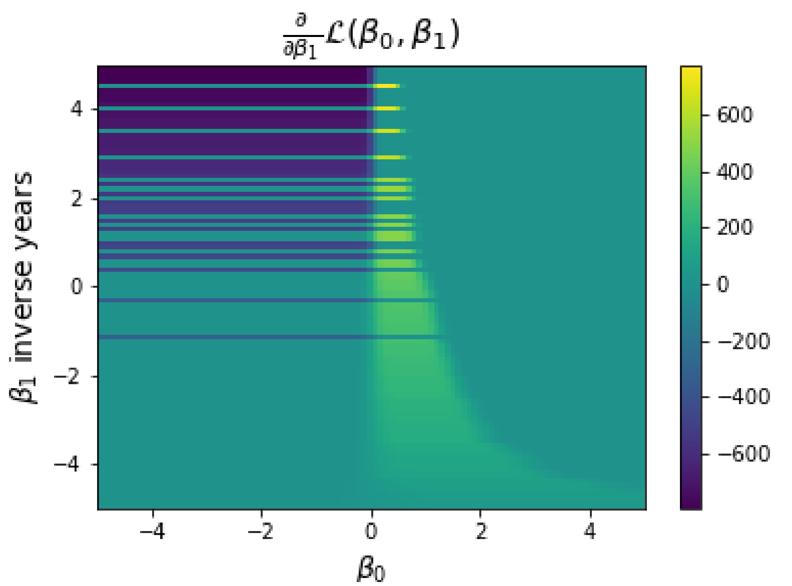
Here is a plot of the log likelihood function as a function of the two parameters.



This plot helps us determine approximate locations of optimal parameters.

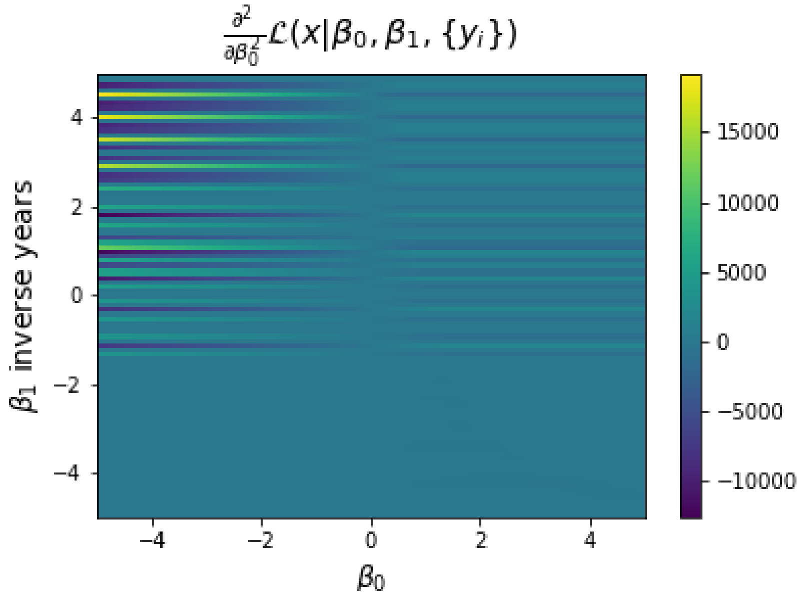
Here are plots of the gradients of the log likelihood function.

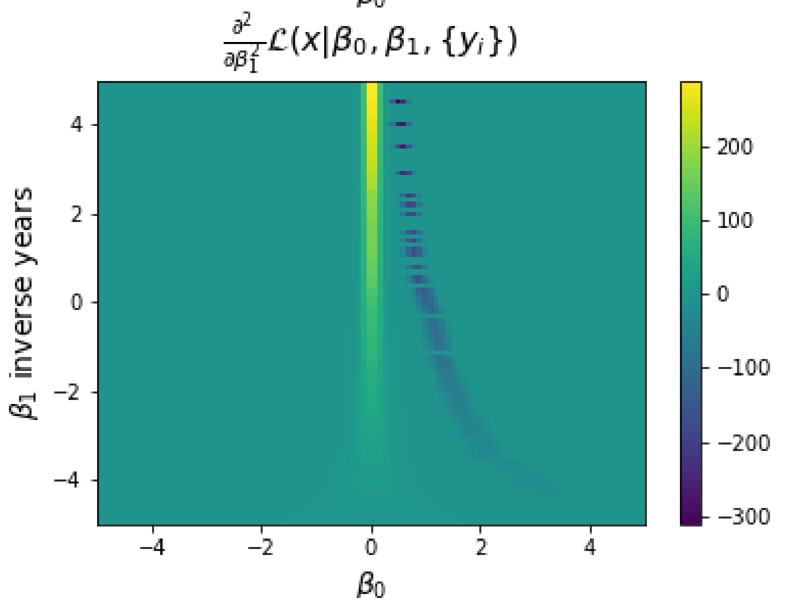


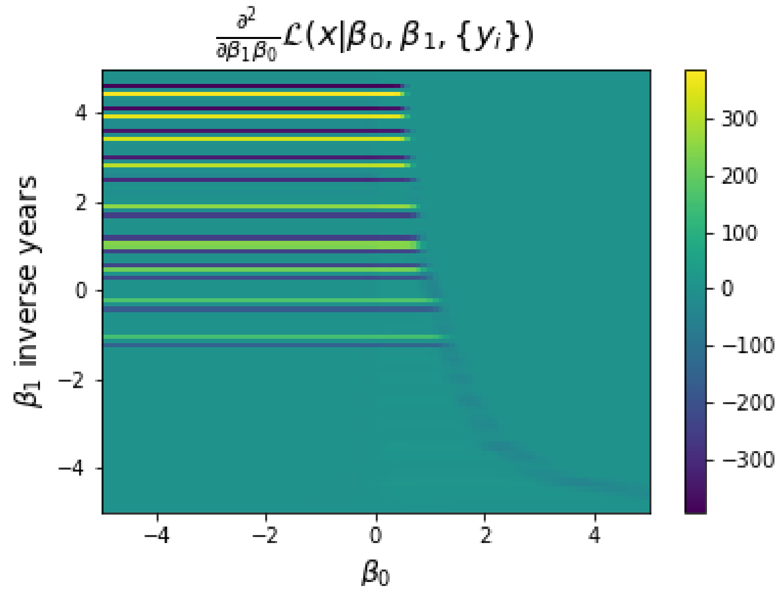


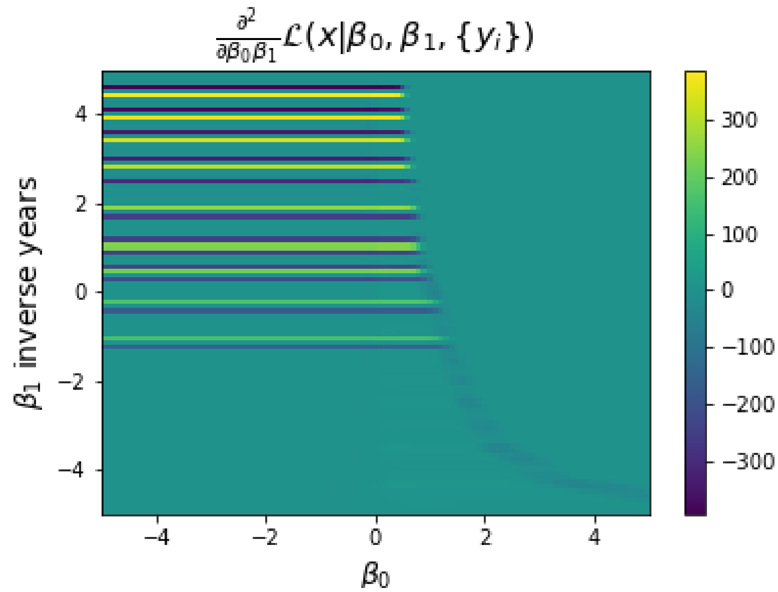
These plots help us determine approximate locations of zeros of the gradient, which can be used to determine the extrema of the log likelihood function.

Finally, here are the plots of the Hessian determinants.


$\frac{\partial^2}{\partial \beta_0^2}\mathcal{L}(x|\beta_0, \beta_1, \{y_i\})$
β_1 inverse years
β_0
−4
−2
0
2
4
15000
10000
5000
0
−5000
−10000


$\frac{\partial^2}{\partial \beta_1^2}\mathcal{L}(x|\beta_0, \beta_1, \{y_i\})$
β_1 inverse years
β_0
−4
−2
0
2
4
200
100
0
−100
−200
−300





The last two plots agree as they should.

Below are the optimal parameters, their errors, and the covariance matrix. We note that the first (second) parameter is β_0 (β_1).

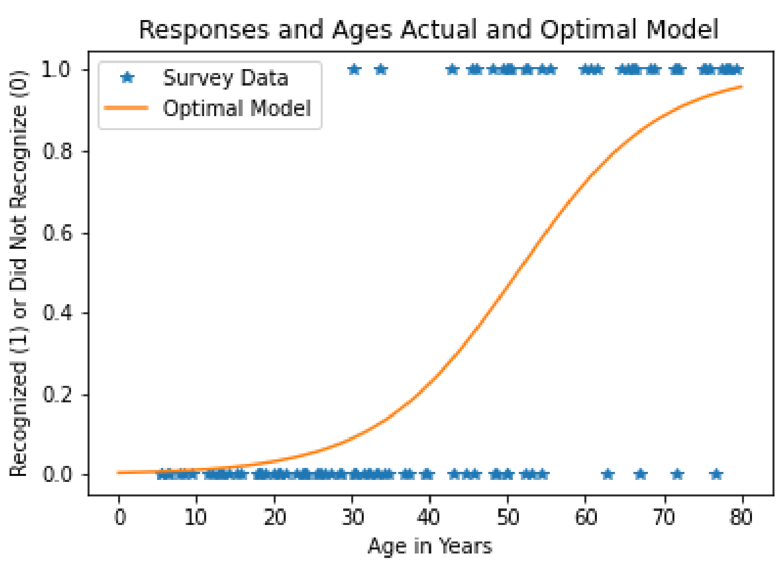
```
Optimal parameters and error:
    p:  [-5.62023158  0.10956337]
    dp:  [0.63392086 0.01246972]
Covariance matrix of optimal parameters:
    C:  [[ 4.01855660e-01 -7.58738080e-03]
 [-7.58738080e-03  1.55494035e-04]]
```

β_0 is about -5.6 ± 0.6 and β_1 is about 0.110 ± 0.012 inverse years

The value of the negative log likelihood function at these optimal parameter values is

```
The optimized negative log likelihood associated with these optimal
parameters is  34.725622486752194
```

Here is the plot of the data and the optimal logistic model.



The optimal model's appearance makes sense. Based on the plot of the survey data, we anticipated that the data had some characteristics of a logistic model and this manifests as an optimal model that captures some of the data's behavior. However, the survey data does not look like a perfect, or close-to-perfect, logistic phenomena and this manifests as an optimal model that does not contain many of the data points near its curve.