Project Report of Assignment 3

ID: *14307130013*
E-mail: 14307130013@fudan.edu.cn

Dec. 9^{th}

1 Fitting exponentials

1.1 Abstract

This report discusses a problem of fitting an exponential model to the given data. The model has the form

$$y = \frac{\beta_1}{\beta_2} e^{-(x-\beta_3)^2/(2\beta_2^2)}$$

and the original dataset consists of 35 observations (x_i, y_i). We compute optima by minimizing the residual sum of squares (RSS). To do this, the BFGS method and the Gauss-Newton method are deployed. Moreover, we study the sensitivity of parameters. We conclude that β_3 has notable infuences on the residual and model if normalized by subtracting the mean of x_i.

1.2 Introduction

As a non-linear regression problem, the fitting task is seen as seeking a minimizer for a non-linear least squares. In this case, we solve

$$\boldsymbol{\beta}_{opt} = \arg\min \frac{1}{2} \sum_{i=1}^{n} [y_i - y(x_i, \boldsymbol{\beta})]^2 \triangleq \arg\min f$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^T$. Descent methods for non-linear optimization are iterative. From an initial guess, a method produces a descending direction p to meet $f(x + \alpha p) < f(x)$, where α is given by line search. Relevant algorithms are shown in **Section 1.4** and attached codes.

We observe the sensitivity of the loss function by plotting contours to compare between two parameters. With regard to the model, we generate 5000 more new point pairs based on $\boldsymbol{\beta}_{opt}$.

1.3 Definitions and theories

1.3.1 Non-linear least squares[1]

Non-linear least squares takes the form of least squares analysis and is used to fit a set of m observations with a non-linear model in n unknown parameters. The basis of the method is to approximate the model by a linear one and to refine the parameters by successive iterations.

1.4 Algorithms

1.4.1 Line search

f stands for the loss function and g is the gradient, or ∇f.

Algorithm 1 Line search based on strong Wolfe condition

$\alpha = 1, \tau = 0.9, c_1 = 10^{-4}, c_2 = 0.9$
while $\alpha > 10^{-6}$ **do**
  if $f(\boldsymbol{\beta}^{(k)} + \alpha p^{(k)}) > f(\boldsymbol{\beta}^{(k)}) + c_1 \alpha {p^{(k)}}^T g(\boldsymbol{\beta}^{(k)})$ **then**
    $\alpha = \tau\alpha$
    continue
  end if
  if $|p^{(k)T} g(\boldsymbol{\beta}^{(k)} + \alpha p^{(k)})| > c_2 |{p^{(k)}}^T g(\boldsymbol{\beta}^{(k)})|$ **then**
    $\alpha = \tau\alpha$
    continue
  end if
  return
end while

1.4.2 The BFGS method

Algorithm 2 The BFGS method

$iter = 0, B^{(0)} = I_n$
while $||g(\boldsymbol{\beta}^{(k)})||_2 > 10^{-6}$ **do**
  $iter = iter + 1$
  $p^{(k)} = -{B^{(k)}}^{-1} g(\boldsymbol{\beta}^{(k)})$
  Obtain α by line search
  $s^{(k)} = \alpha p, \boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + s^{(k)}$
  $y^{(k)} = g(\boldsymbol{\beta}^{(k+1)}) - g(\boldsymbol{\beta}^{(k)})$
  $B^{(k+1)} = B^{(k)} + \frac{y^{(k)} {y^{(k)}}^T}{{y^{(k)}}^T s^{(k)}} - \frac{B^{(k)} s^{(k)} {s^{(k)}}^T B^{(k)}}{{s^{(k)}}^T B^{(k)} s^{(k)}}$
end while

[1] From Wikipedia: Non-linear least squares. (https://en.wikipedia.org/wiki/Non-linear_least_squares)

1.4.3 The Gauss-Newton method

J is the Jacobian which is used to approximate the Hessian H that $H \approx J^T J$.

Algorithm 3 The Gauss-Newton method

$iter = 0$
while $||g(\boldsymbol{\beta}^{(k)})||_2 > 10^{-6}$ **do**
  $iter = iter + 1$
  $p^{(k)} = -[J(\boldsymbol{\beta}^{(k)})^T J(\boldsymbol{\beta}^{(k)})]^{-1} g(\boldsymbol{\beta}^{(k)})$
  Obtain α by line search
  $\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \alpha p$
end while

1.5 Implementation issues discussion

1.5.1 Model parameters

Starting from (1, 3, 450), the two methods converges at $iter = 5$ (Gauss-Newton) and $iter = 17$ (BFGS). Both provides nearly equal optima $\boldsymbol{\beta}_{opt} = (1.5544, 4.0888, 451.5412)$ with the final loss $7.3179e - 04$.

1.5.2 Sensitivity study

We observe the sensitivity of each parameter by fixing one or two of them and see the other(s) as variable(s). Here we scale down β_3 by 450, i.e. $\bar{x}$, so that three parameters are at the same order of magnitude. We plot contours as well as curves in Figure 1 to Figure 6.

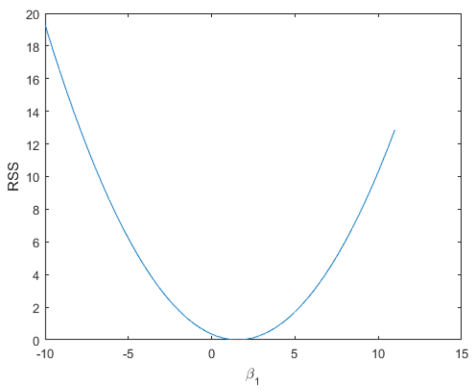

Figure 1: Fix β_2, β_3

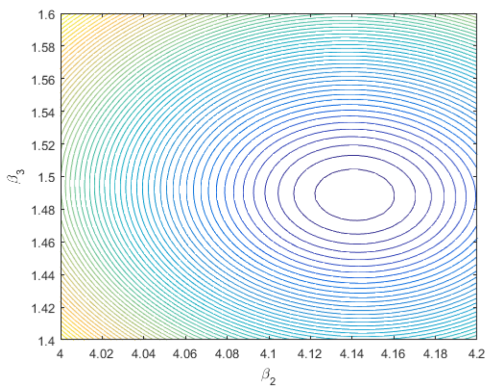

Figure 2: Fix β_1

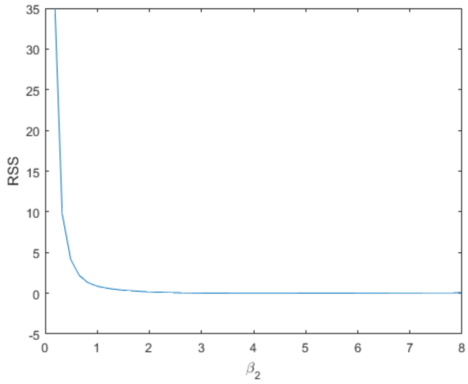

Figure 3: Fix β_1, β_3

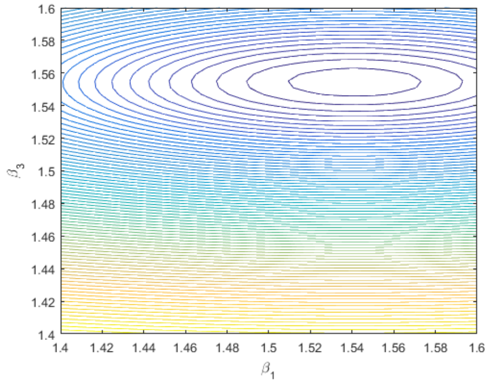

Figure 4: Fix β_2

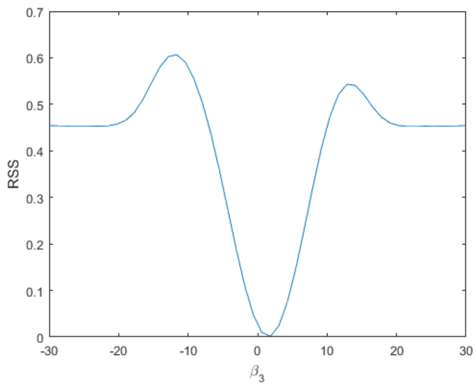

Figure 5: Fix β_1, β_2

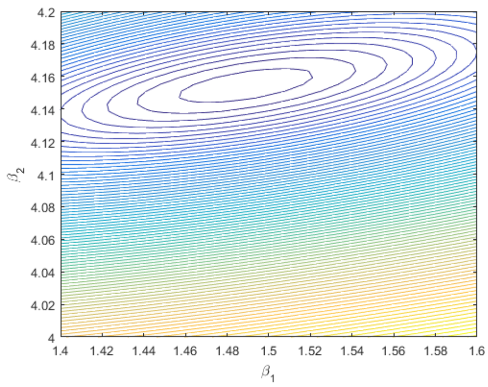

Figure 6: Fix β_3

The loss function is (locally) the most sensitive to β_3, followed by β_2 and β_1, as slopes or the density of contour lines suggest. This is because β_1 is the linear parameter, which does not significantly affect the sensitivity. From the right row of the plots, we can see that when one parameter is fixed, there is a wide range of values of the other two producing small residuals.
As to the model, we use $\boldsymbol{\beta}_{opt}$ to generate new data out of $x^{(new)} \in (\min(x_i), \max(x_i))$ with noise from $\mathbb{N}(0, 0.1)$. After scaling, we have $\boldsymbol{\beta}_{opt} = (1.5544, 4.0888, 1.5412)$. The average changing rates/variances (20 runs) are $0.43\%/3.01e{-}04$, $1.15\%/3.44e{-}04$ and $0.32\%/0.0035$. On the other hand, β_1 has the highest changing rate and variance if β_3 is not normalized.

1.6 Experiment results

1.6.1 Remarks on two algorithms

Both algorithms avoid computing the Hessian. For BFGS, since the dimension of the target is only three, the memory problem does not occur. It has a slower convergence speed due to the initial B.
The Gauss-Newton method takes fewer iterations but there may be divergence if the initial β_3 is far away from the $\bar{x}$ (as shown in Figure 5).

1.6.2 Discussion on sensitivity

The change of β_3 is more frequent than β_1 and β_2 as it has the largest variance. That is to say once some noise is introduced, β_3 is the most sensitive.

1.7 Conclusion

We solve a non-linear least squares problem through the BFGS method and the Gauss-Newton method. The magnitude of error is around 10^{-4}.
From the density of contours, we know that the loss function is quite sensitive to β_3.
From the respective of the model, the order of magnitude has a lot to do with the sensitivity of parameters.

1.8 Acknowledgement

Great gratitude to TA Lu for providing an evaluation method of sensititvity.

1.9 References

1. https://en.wikipedia.org/wiki/Non-linear_least_squares
2. https://en.wikipedia.org/wiki/Broyden-Fletcher-Goldfarb-Shanno_algorithm
3. https://en.wikipedia.org/wiki/Gauss-Newton_algorithm