Project Report: Finding Maximum Likelihood Estimators using the Generalized Newton's Method

Yunus Tuncbilek

Due May 5, 2016

Abstract

This report demonstrates an approach to Newton's method, generalizing the method to provide new routines that take less number of iterations and fail less. Newton's method could actually be thought of as iteratively maximizing a local quadratic approximation to an objective function. This approach is studied in [4] and [5]. In this report, we apply non-quadratic local approximations using various class $C^{k\geq 2}$ functions and find their maximum. We iteratively update the x-value and approach to the maximum of the objective function. Then, we compare the results of this new optimization routine to the other ones, mainly to Newton's method.

1 Motivation: A Geometric Interpretation of Newton's Method using Taylor's Theorem

The primary motivation behind this project stems from Taylor's Theorem:

Suppose g is a function $\mathbb{R}^n \to \mathbb{R}$ of class C^k function on an open convex set S. Then, at every point a in S, g could be approximated by a polynomial $h(x)$ of degree at most k [1]. In the context of optimization, Taylor's Theorem could be utilized by iteratively generating local smooth approximations to an objective function and using the maxima of these functions to get step sizes. In particular, Newton's method is nothing more than the quadratic case of this phenomena.

Let f be a twice continuously differentiable $\mathbb{R}^n \to \mathbb{R}$ function and solve

$$\min_x f(x)$$

At $x = \bar{x}$, by Taylor's Theorem, we know that

$$f(x) \approx h(x) := f(\bar{x}) + \nabla f(\bar{x})^T(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T H(\bar{x})(x - \bar{x})$$

where H is the Hessian of f. Note that $h(x)$ is maximized by solving $\nabla h(x) = 0 \implies \nabla h(x^*) = \nabla f(\bar{x}) + H(\bar{x})(x^* - \bar{x}) = 0$. Thus, assuming that $H(\bar{x})$ is invertible, the above equation yields

$$x^* = \bar{x} - H(\bar{x})^{-1}\nabla f(\bar{x})$$

The direction $-H(\bar{x})^{-1}\nabla f(\bar{x})$ is called the Newton step at $\bar{x}$, leading to Algorithm 1, which is called Newton's Method in optimization. We know that if $H(x)$ is positively semi-definite and the direction

Algorithm 1: Newton's Method

Input : A starting point $x^{initial}$
while $d^k \neq 0$ **do**
  $d^{k+1} = -H(x^k)^{-1}\nabla f(x^k)$;
  $x^{k+1} \leftarrow x^k + d^{k+1}$;
end

is nonzero, then d^k is a descent direction [2].

Therefore, given that the Hessian is positively semi-definite, the geometric interpretation of Newton's method could be that at each iteration one approximates $f(x)$ by a quadratic function around x^k, and then takes a step towards the maximum/minimum of that quadratic function (note that, although it is easy to visualize it in lower dimensions, in higher dimensions this may also be a saddle point).

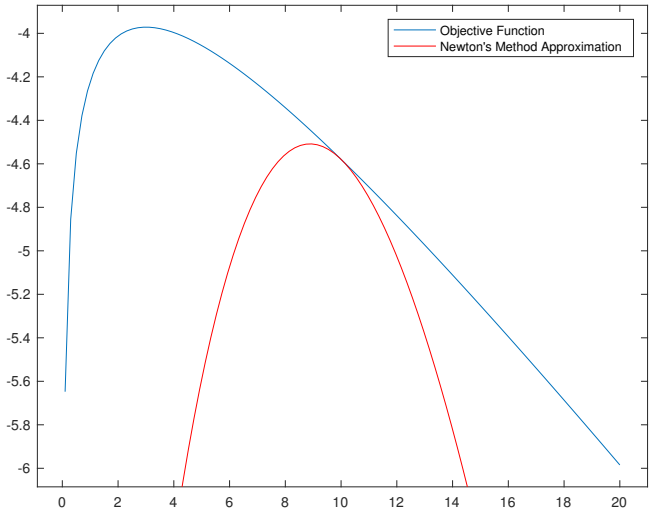


Figure 1: The figure shows the quadratic function that Newton's method starting at $x^{initial} = 10$ generates in the first step. The labels should be x and the log likelihood of x for the dataset. The dataset was sampled from the Gamma distribution with parameters $a = 3$ and $b = 8.5$, and the objective function shows its probability density function.

However, we realize that this quadratic approximation does not fit the function very well and produce a slow convergence rate. Instead, we could use a logarithmic function that fits it better, producing better step sizes, as shown in Figure 2. In this project, the goal is to find functions that suit the objective better and apply the new Newton's step to find the maximum/minimum faster and more robustly.

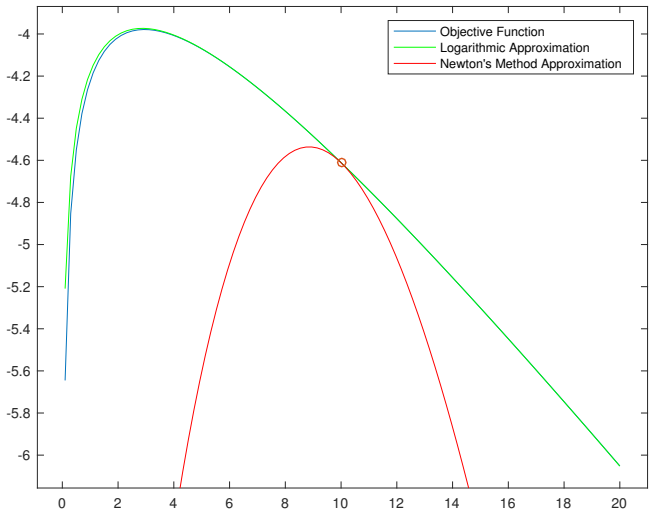


Figure 2: Here we use a local approximation function in the form $c_0 + c_1 log(x) + c_2 x$, which fits the function much better (the same Gamma distribution was used)

2 Optimization Problem: Maximizing Likelihood Functions

In this report, we focus on the likelihood functions of the parameters of various distributions.

Consider a random vector Y, and assume that we have N i.i.d. observations from this vector. Denote $f(Y|\theta)$ the probability density function of Y with respect to a set of parameters θ. For the scope of this report, we will assume that Y is continuous. Since the observations are independent, we could write the joint density function as [3]

$$f(y_1, y_2, \dots y_n|\theta) = f(y_1|\theta)f(y_2|\theta)\dots f(y_n|\theta)$$

Since the observations are given, we are not really interested in them, but rather, we want to find good estimates of the parameters, θ. For this purpose, the likelihood function is defined as follows

$$L(\theta|y_1, y_2, \dots y_n) = f(y_1, y_2, \dots y_n|\theta) = f(y_1|\theta)f(y_2|\theta)\dots f(y_n|\theta)$$

The unconstrained optimization problems that we deal with in this project are of the form

$$\max_{\theta\in\Theta} L(\theta|y_1, y_2, \dots y_n)$$

where θ is being optimized and the likelihood function is the objective function. We define the solutions to these problems as maximum likelihood estimators

$$\hat{\theta}_{MLE} = \operatorname*{argmax}_{\theta\in\Theta} L(\theta|y_1, y_2, \dots y_n)$$

The natural logarithm of the likelihood function is defined as the log-likelihood function. Note that the value of the likelihood function is greater than zero, so the solutions to their maximization problems (MLEs) would be the same. We mostly look at $\mathbb{R} \to \mathbb{R}$ likelihood or log-likelihood functions and estimate one of the parameters, taking in the others as given.

3 Example: Gamma Distribution

A random variable follows the gamma distribution with shape paramater $a > 0$ and scale parameter $b > 0$ (meaning that $\theta = a, b$ if it has the probability density function f given by

$$f(x|a,b) = \frac{x^{a-1}}{\Gamma(a)b^a}e^{-\frac{x}{b}}$$

As given in [**?**], the log-likelihood function for the sample $[X_i]_{i=1}^n$ is

$$\log L(a,b|X_i) = (a-1)\sum_{i=1}^{n}\log X_i - n\log\Gamma(a) - na\log b - \frac{1}{b}\sum_{i=1}^{n}$$

Since $\frac{\partial log(L(a,b|X_i))}{\partial b} = 0 \implies b^* = \frac{\sum_{i=1}^n x_i}{a}$, we can easily optimize the log-likelihood with respect to b. Plugging it in, we get (using the bar notation for the sums as in [4])

$$\log L(a,b^*|X_i) = n(a-1)\overline{\log x} - n\log\Gamma(a) - na\log\bar{x} + na(\log a - 1)$$

Therefore, we want to solve the following unconstrained maximization problem

$$\max_a \left(n(a-1)\overline{\log x} - n\log\Gamma(a) - na\log\bar{x} + na(\log a - 1)\right)$$

We will look at three algorithms to maximize this function. Note that the second derivative of the function is $-n\Psi_1(a) + \frac{n}{a} < 0$ for all a, where Ψ_1 is the trigamma function. So, it is concave.

The first method will iteratively maximize a linear lower bound [4].Since $a\log a$ is convex (first derivative is $1 + \log a$ and the second derivate is positive $\frac{1}{a}$), if we pick a starting point a_0

$$a\log a \geq (1+\log a_0)(a - a_0) + a_0\log a_0$$

$$\implies \log L(a,b^*|X_i) \geq n(a-1)\overline{\log x} - n\log\Gamma(a) - na\log\bar{x} + n(1+\log a_0)(a-a_0) + na_0\log a_0 - na$$

Therefore the maximum of the lower bound happens at $\Psi(a^*) = \overline{\log x} - \log\bar{x} + \log a_0$, where Ψ is the digamma function [4]. We showed that the initial likelihood function is log-concave, so the lower

Algorithm 2: Lower bound algorithm

Input : A starting point $x^{initial}$
while $x^k \neq x^{k-1}$ **do**
  $x^{k+1} \leftarrow \Psi^{-1}(\overline{\log x} - \log\bar{x} + \log x_k)$;
end

bound algorithm would converge to the global maximum. Minka argues, the lower bound algorithm takes around 250 iterations to reach the maximum [4]. I implemented Algorithm 2 on Matlab with stopping condition $|x^k - x^{k-1}| < 10^{-8}$ and starting point $a = 10$ (as in Minka's paper). I found that

it requires 292 iterations to reach the maximum and is very inefficient. As shown in Figure 3, it takes very short steps towards the end.

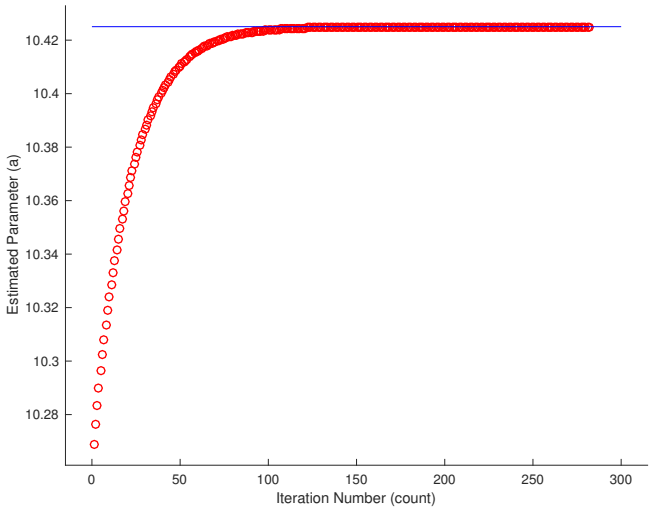


Figure 3: The convergence of lower bound algorithm with $a = 10$

We fix this inefficiency by using Newton's method. As described in the first chapter, Newton's method would have step size $\frac{-f'(a)}{f''(a)}$, leading to the following algorithm (here we divide $\log L(a, b^*|X_i)$ by n to simplify the computation and denote the new function as f)

Algorithm 3: Newton's algorithm

Input : A starting point $x^{initial}$
while $x^k \neq x^{k-1}$ **do**
  $f'(x^k) = log(x^k) - s - \Psi(x^k)$;
  $f''(x^k) = -\Psi_1(x^k) + 1/x^k$;
  $x^{k+1} \leftarrow x^k - \frac{f'(x^k)}{f''(x^k)}$;
end

where s is equal to $\overline{\log x} - \log \overline{x}$.

We could also use the heuristic from the first chapter and locally approximate the objective iteratively by using a function of type $c_0 + c_1 log(a) + c_2 a$ [4]. Therefore, our approximation has the form

$$g(a) = c_0 + c_1 log(a) + c_2 a$$

$$g'(a) = \frac{c_1}{a} + c_2$$

$$g''(a) = \frac{-c_1}{a^2}$$

Matching its derivatives and value with those of $\log L(a, b^*|X_i)$, we calculate c_1 and c_2 as follows

$$c_1 = \left(\frac{1}{a} - \Psi_1(a)\right) a^2$$

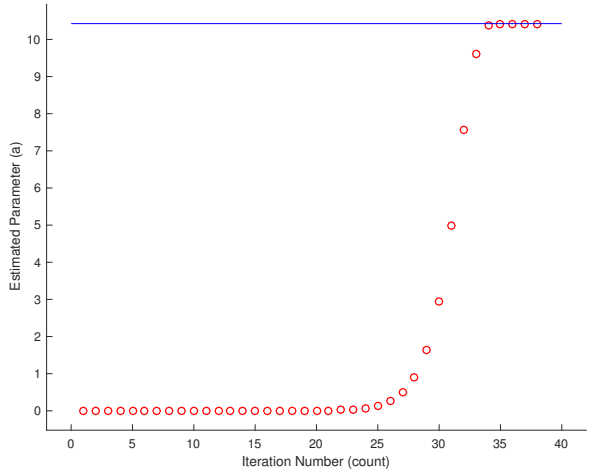
$$c_2 = \overline{\log x} - \log \bar{x} + \log a - 1 + a\Psi_1 a$$

Since the local approximation is maximized at $\frac{-c_2}{c_1}$, the resulting algorithm is the following.

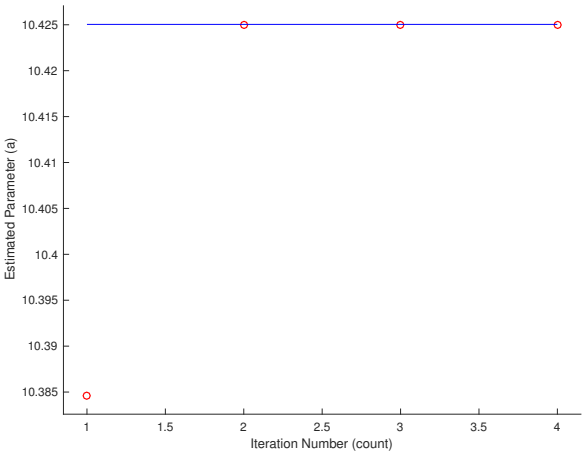
Algorithm 4: Generalized Newton's Algorithm

Input : A starting point $x^{initial}$
while $x^k \neq x^{k-1}$ **do**
  $c_1 = \left(\frac{1}{x^k} - \Psi_1(x^k)\right) {x^k}^2$;
  $c_2 = \overline{\log x} - \log \bar{x} + \log x^k - c_1/x^k$;
  $x^{k+1} \leftarrow -\frac{c_2}{c_1}$;
end

Although Minka does not mention the implementation of Newton's method in Gamma distributions, I found that this algorithm (Algorithm 4) fits the objective much better than Newton's method (Figure 1 for example) and converges faster than Newton's regardless of the starting point. Figure 4 shows an example with samples $X = \{1, 2, 2\}$, stopping condition $|x^k - x^{k-1}| < 10^{-8}$ and starting point $a_0 = 20.7692$.



(a) Newton's Algorithm



(b) Generalized Newton's Method (Algorithm 4)

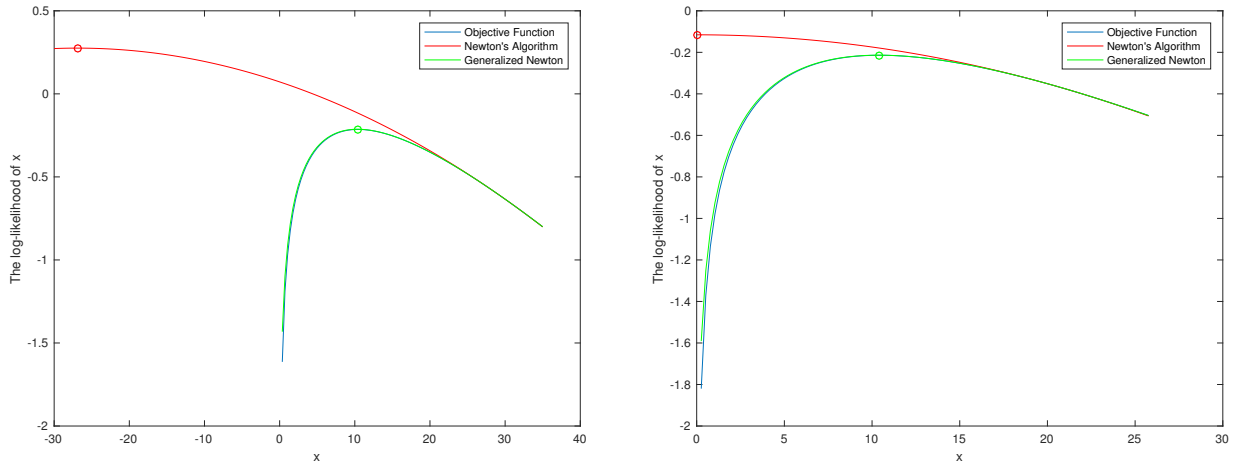
Figure 4: The convergence of Newton's Algorithm and Generalized Newton's Algorithm (Algorithm 4)

As we can see from the figure, Newton's algorithm indeed takes smaller steps towards the end compared to the lower bound algorithm. However, it takes very short steps at the beginning, requiring

39 steps to reach the maximum, while Algorithm 4 only requires 4. Algorithm 4 makes an efficient jump at the second step as we can see from the following table.

Iteration Number	Newton	Algorithm 4
0	20.7692	20.7692
1	1.0006e-08	10.3846
2	2.0012e-08	10.4250
3	4.0023e-08	10.4250

We also realize that Newton's method jumps to almost 0, which is the boundary point of the domain, implying that any starting point significantly larger than $a_0 = 20.7692$ would cause Newton's method to fail.[1] This shows that the Generalized Newton's Method is not only efficient but also robust. An example of Newton's failure is shown below with starting point $a_0 = 30$.



(a) The case when $a_0 = 30$

(b) The previous case when $a_0 = 20.7692$.

Figure 5: The figure shows the approximating functions in the first step. The red and green circles are the updated values in Newton's Algorithm and Algorithm 4, respectively.

The reason why Newton takes a large step and goes out of the domain is that the second derivative of the log likelihood function goes to zero as x gets larger, which is shown in Figure 6. This will be discussed further in the next chapters.

[1] In fact, I purposely chose this point by solving the equation

$$a + \frac{\Psi(a) + \overline{\log x} - \log \overline{x} - \log a}{-\Psi_1(a) + \frac{1}{a}} = 0$$

so that Newton's algorithm would have to take many steps at the beginning. So, we can choose a better estimate of the solution to this equation and use it as the starting point. This way, we could make Newton's method require an even higher number of steps.

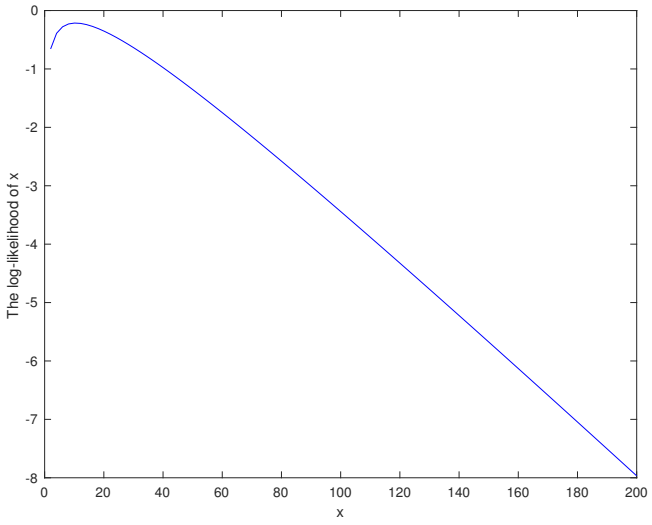


Figure 6: This is the graph of the Gamma log-likelihood from sample $X = [1, 2, 2]$. The second derivative approaches to zero as x goes to infinity

4 Example: Lévy Distribution

Lévy distribution with location parameter $\mu = 0$ is given by the following probability density function

$$f(x|c, \mu = 0) = \sqrt{\frac{c}{2\pi}} \frac{\exp(-\frac{c}{2x})}{x^{3/2}}$$

Thus the log-likelihood is

$$\log L(c|x, \mu = 0) = \frac{n}{2} \log \frac{c}{2\pi} - \sum_{i=1}^{n} \frac{c}{2x_i} - \frac{3}{2} \sum_{i=1}^{n} \log x_i$$

which is in the form $g(c) = k_0 + k_1 c + k_2 \log c$ where k_i's are constant real numbers. Thus, applying the generalized Newton's method with the approximation that we used in the previous chapter for Algorithm 4, we can indeed get to the maximum likelihood estimator of Lévy Distribution in just one step. [2] On the other hand, Newton's method still takes about 6 iterations to find the MLE and sometimes fails. The following table shows the first four iterations of the algorithms generated by Newton's method and the generalized Newton's method, when $x = \{1, 2, 2, 2, 3, 12, 3, 1, 1\}$ and the starting point $c_0 = 1$.

Iteration Number	Newton	Generalized Newton
0	1.0000	1.0000
1	1.6626	1.7143
2	1.7127	MLE found
3	1.7140	MLE found

[2] In fact, the MLE for c is equal to $\frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$, which is what the first iteration of the generalized Newton's Method with a function of the form $g(c) = k_0 + k_1 c + k_2 \log c$ generates

5 Example: Cauchy Distribution

The probability density function of Cauchy distribution is given by

$$f(x|\gamma, x_0) = \frac{\gamma}{\pi(1 + (\gamma(x - x_0))^2}$$

Therefore, setting the location parameter $x_0 = 0$, we get the following log-likelihood function for the shape parameter γ. [3]

$$\log L(\gamma|x, x_0 = 0) = l(\gamma) = n \log \gamma - n \log \pi - \sum_{i=1}^{n} \log(1 + \gamma^2 x_i^2)$$

Its derivatives are

$$l'(\gamma) = \frac{n}{\gamma} - \sum_{i=1}^{n} \frac{2\gamma x_i^2}{1 + \gamma^2 x_i^2}$$

$$l''(\gamma) = \frac{-n}{\gamma^2} + \sum_{i=1}^{n} \frac{2x_i^2 - 2\gamma^2 x_i^4}{(1 + \gamma^2 x_i^2)^2}$$

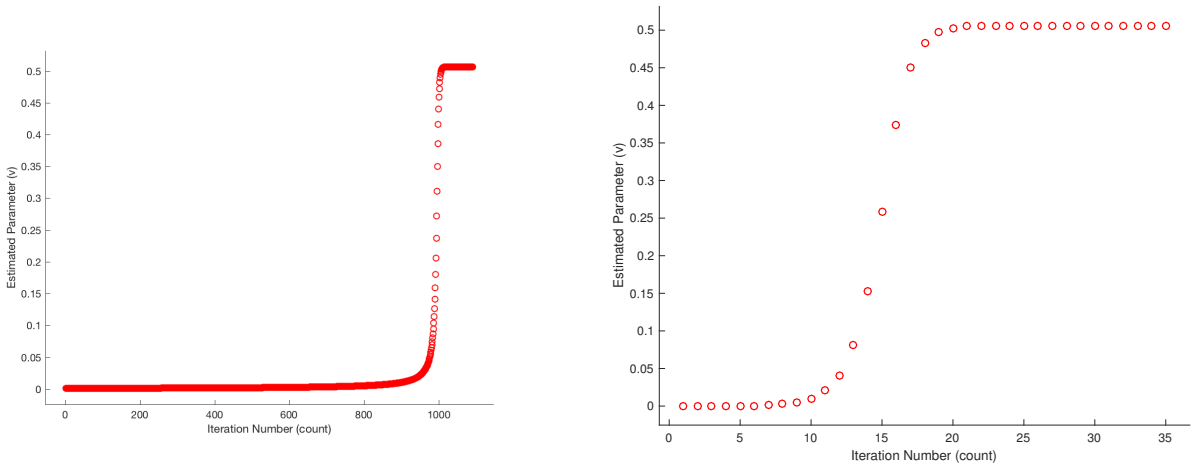
We choose to apply the generalized Newton's method with a function of the form $g(\gamma) = c_0 + c_1 \log(\gamma + 1) + c_2 \log(\gamma)$. By matching its derivatives with those of f, we get the following algorithm

Algorithm 5: Generalized Newton's Algorithm for Cauchy Distribution

Input : A starting point $x^{initial}$
while $\gamma^k \neq \gamma^{k-1}$ **do**
  $c_2 = -\gamma^k l''(\gamma^k)(\gamma^k + 1) - \gamma^k l'(\gamma^k)$;
  $c_1 = (l'(\gamma^k) - (c_2/\gamma^k))(\gamma + 1)$;
  $x^{k+1} \leftarrow -\frac{c_2}{c_1 + c_2}$;
end

Unlike the previous applications with Lévy and Gamma distributions, generalized Netwon's method does not produce a better algorithm than Newton's method.

[3] Minka applies the generalized Newton's method to estimate the location parameter [5]



(a) Generalized Newton's Method (Algorithm 5)

(b) Newton's Algorithm

Figure 7: The figure shows the convergence of the two algorithms when the starting point $\gamma_0 = 0.001$ and the sample is $X = \{1, 2, 2, 2, 3, 12, 3, 1, 1\}$

As seen in the above figure, Algorithm 5 takes about 1000 iterations when the starting point is close to zero, while Newton's algorithms requires around 40. If we take a closer look at what happens at the first step, we get the following figure.

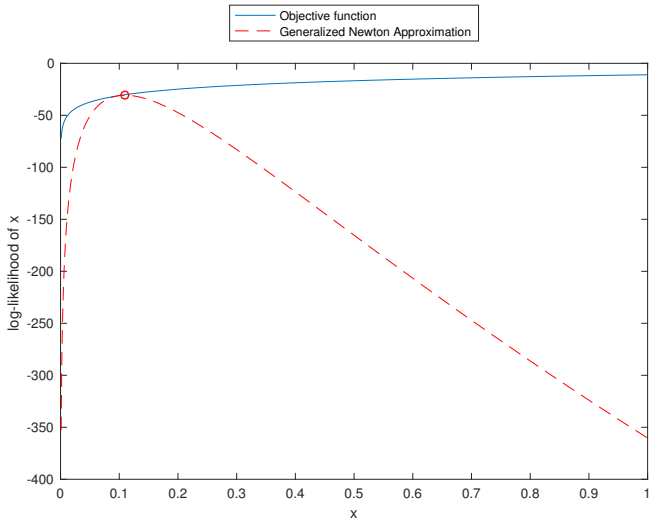


Figure 8: This is the graph of the Cauchy log-likelihood from the previous sample and with starting point $\gamma_0 = 0.1$ and the approximating function that Algorithm 5 produces in the first step

In Figure 8, we see that the functions that Algorithm 5 produces do not fit the log-likelihood function as perfectly as it does in the previous examples with Lévy and Gamma distributions. This discrepancy could lead to extremely bad results for Algorithm 5. For instance, when the starting point is $v_0 = 10^{-6}$ and the data sample is X (defined earlier), Algorithm 5 requires more than a million

iterations to reach the maximum with the stopping condition $|\gamma^k - \gamma^{k-1}| < 10^{-8}$.

6 Comments

In chapters 3 and 4, the generalized Newton's method turned out to be very successful. It solved the maximum likelihood estimation problems of Gamma and Lévy both efficiently and robustly, meaning that it always took less number of steps than Newton's method and it did not depend much on the starting point. With Lévy distribution, we had a special case because the local approximation function used in Chapter 3 could find the maximum of the Lévy distribution in one step.

At the end of Chapter 3, I observed that when the second derivative of the objective function is too small (goes to zero), the generalized Newton's method performs much better than Newton's method. The step size of the generalized Newton algorithm in chapter 3 is

$$\Delta x = \frac{-xf'(x)}{f'(x) + f''(x)x}$$

while the step size of Newton is $\frac{-f'(x)}{f''(x)}$. So, when $f''(x)$ is too small, Newton takes unnecessarily big step like it does in Figure 5. On the other hand, since generalized Newton is a good approximation of the function, it takes reasonably good first steps as we see in the table on page 7.

I found implementing the algorithms to be a lot of fun. For each algorithm, I wrote a Matlab program to run them and generate their graphs. I used functions like digamma(), trigamma() etc. from the Lightspeed Matlab Toolbox built by Minka [6]. Some of the code in the toolbox was not up-to-date. I had to edit the file install-lightspeed.m so that it could work with the new version of Matlab.

Excitingly, my Matlab program that finds Gamma MLEs takes much less time than the actual gamfit() function in Matlab. When I run the following code

```
1 for iter = 1:10
2     a = mygammafit(repmat([iter iter+1 iter+2], 1, 100000000))
3 end
```

and a similar code for Matlab's gamfit() function, the latter takes about 45s, while the former takes less than 20s.

In chapter 5, we try estimating the scale parameter of a Cauchy distribution, but cannot compete with Newton's method. In Figure 8, we see that the approximation function of the generalized Newton method does not fit the objective function well. We could have chosen a more suitable function like $g(\gamma) = c_0 + c_1 \log(c_4\gamma^2 + 1) + c_2 \log(\gamma)$, but, with four c_i's, the system of equations is very complicated and it is hard to get the updated c_i's. Honestly, even though I knew that this function did not exactly resemble the log-likelihood of Cauchy, I was not expecting the quadratic approximation to perform better because the objective function was not quadratic at all. However, such intriguing unexpected behaviors made my experience much more interesting.

7 Summary

In brief, this project compares the generalized Newton's method and the conventional Newton's method, building on the heuristic introduced in [5]. The choice of approximating functions is almost up to us, although we could give some guidelines. Choosing resembling terms helps the optimization as we have seen in chapters 3 and 4. If the second derivative of the objective function is too small, using the function in Chapter 3 would be helpful. The complexity of the calculations needed to get the coefficients constrains our choice (as it does in chapter 5). In those cases, Newton's method could a better option.

In the future, we should conduct a statistical analysis to see if a function would fit the objective well without implementing the respective algorithms. This analysis could also lead to finding optimal approximating functions to estimate the arg max of certain log-likelihoods. In this project, we only estimate one of the parameters, but we could also study maximum likelihood estimators with two or more parameters and come up with multivariable approximating functions. Lastly, we do not discuss the optimal choice for the starting point, so in the future research, we should formulate this choice as well.

References

[1] Folland, G. B. "Higher-Order Derivatives and Taylor's Formula in Several Variables." (2005).

[2] Freund, Robert M. "Newton's Method for Unconstrained Optimization." (2004).

[3] Mai, Anh Tien, Fabian Bastin, and Michel Toulouse. On Optimization Algorithms for Maximum Likelihood Estimation. No. 64. CIRRELT Technical Report, 2014.

[4] Minka, Thomas P. "Estimating a Gamma distribution." Microsoft Research, Cambridge, UK, Tech. Rep (2002).

[5] Minka, Thomas P. "Beyond newton's method." (2000).

[6] Minka, Tom. "The Lightspeed Matlab toolbox." Efficient operations for Matlab programming, Version 2 (2003) url: http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/.