Quadratic Approximation, Fixed Step Size and Armijo Backtracking Search

Matthew Sun

1 Abstract

The purpose of this assignment was to estimate a minimizer for three differentiable functions that were locally convex at the observed region using one step of a quadratic approximation given one point, fixed step size line search, and line search with Armijo backtracking. The three objective functions to be estimated and their respective brackets that define the observed region were $f_1(t) = e^{(3t)} + 5e^{(-2t)}$ with bracket $[0, 1]$, $f_2(t) = \ln(t)^2 - 2 + \ln(10-t)^2 - t^{0.2}$ with bracket $[6, 9.9]$, $f_3(t) = -3t\sin(0.75t) + e^{(-2t)}$ with bracket $[0, 2\pi]$. It was found that this implementation of quadratic approximation performed poorly compared to the two line search algorithms. The estimated minimizer differed by 0.438 for f_1, 1.206 for f_2, and 3.860 for f_3 from the actual minimizer. Both line search algorithms managed to converge and provided very accurate estimates that when rounded to three decimal places were virtually identical to the actual minimizer. It was also found that Armijo backtracking significantly reduced the number of iterations until convergence in comparison to the fixed step search for this implementation applied to these functions.

2 Introduction

In this assignment, three methods of approximations were used to find the local minimizer of three separate differentiable functions with scalar inputs that are locally convex at a given subset of input values. The methods used were 1. quadratic approximation using a single point plus the first and second derivative at that point, 2. line search with a fixed step size, and 3. line search with dynamically estimated step sizes using Armijo backtracking. These methods were assessed based on whether they converged for the given termination conditions and if they did, how close they were to the actual minimizer as well as how many iterations it took to get there. There is no iteration involved in quadratic approximation so the model will only be evaluated based on its proximity to the actual minimizer.

2.1 Quadratic Approximation

Quadratic approximation is the process by which a second-order polynomial is used to estimate the stationary point(s) in a local region of an often more complex function. A stationary point for any function $f : \mathbb{R} \to \mathbb{R}$ is defined as t^* for any $t^* \in \mathbb{R}$ such that $f'(t^*) = 0$. Note that although a minimizer is necessarily a stationary point, it is not the only requirement. Let the second-order polynomial model for an objective function $f(t)$ be $p(t) = a_1t^2 + a_2t + a_3$. To acquire the coefficients for this model a system of three equations must be constructed. The system of equations for a model that is provided with three points t_1, t_2, t_3 as a Vandermonde matrix is

$$\begin{bmatrix} t_1^2 & t_1 & 1 \\ t_2^2 & t_2 & 1 \\ t_3^2 & t_3 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(t_1) \\ f(t_2) \\ f(t_3) \end{bmatrix} \tag{1}$$

In the case where the model is only provided with one point, the Vandermonde matrix will reflect the fact that the inputs will be $f(t_1), f'(t_1), f''(t_1)$ and two of the rows will become the first and second derivative of $p(t)$. The Vandermonde matrix for a quadratic approximation with one point provided is

$$\begin{bmatrix} t_1^2 & t_1 & 1 \\ 2t_1 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(t_1) \\ f'(t_1) \\ f''(t_1) \end{bmatrix} \tag{2}$$

The minimizer $\hat{t}$ of this quadratic model is found at the point where $p'(\hat{t}) = 2a_1\hat{t} + a_2 = 0$.

The scientific question to be explored is: Will a quadratic approximation for each of the given functions be able to give an accurate estimate of the actual local minimizer and how will it compare to the other methods? The effectiveness of the quadratic model was assessed based on the difference between the estimated minimizer and the actual local minimizer.

2.2 Line Search and Armijo Backtracking

Line search is a process of successive iterations for estimating the local minimizer for a function. A starting estimate and step size are required for the fixed step size algorithm. Given a function $f(t)$, each iteration of a fixed step size search, the new estimate t_{k+1} for iteration k is

$$t_{k+1} = t_k + s_0 d_k \tag{3}$$

where s_0 is the inputted step size and d_k is the directional derivative for that iteration and is defined as $d_k := -f'(t_k)$. Iterations would be repeated until the estimate converges or reaches a maximum number of allowed iterations in practice.

Choosing an appropriate step size is a major concern for the fixed step size search as too large a step size can cause the estimates to oscillate between values and miss the local minimizer entirely. Conversely, too small a step size makes the algorithm inefficient by increasing the number of iterations required for convergence. This problem can be mitigated by estimating an appropriate step size for each iteration separately, such a process is called backtracking. One method is exponential back-off where, starting at iteration $\gamma = 0$, increment γ for the line search iteration k until $f(t_k + \beta^\gamma s_0 d_k) < f(t_k)$ then stop where β is the backtrack rate. An assumption that is made about backtracking is that the user inputted step size s_0 is too large and backtracking estimates a better step size by decreasing the step size by a factor of β until the newly estimated step size results in an improvement (decrease in the objective function value). Exponential back-off can be improved upon using Armijo backtracking [Arm66] by adding an extra condition to the right-hand side of the previously described condition. So similarly, γ is incremented until $f(t_k + \beta^\gamma s_0 d_k) < f(t_k) + \alpha_k \beta^\gamma s_0 d_k$ where $\alpha_k := f'(t_k)/2$. The added component, known as the Armijo condition, works by exponentially decaying the magnitude of the slope of the line at the current $f(t_k)$. One can imagine that as γ increases, this line becomes increasingly flat and eventually approaches that of the exponential back-off.

The scientific question to be explored is: Will the fixed step size line search and Armijo backtracking search converge and if they do how close are each of the estimations to the actual local minimizer, how many iterations did it take to reach the output, and how do the algorithms compare to each other and quadratic approximation? The results were evaluated based on whether each algorithm converged and the difference between the actual local minimizer and the estimated as well as the number of iterations it took to get there.

3 Methods

The three objective functions to be estimated and their respective brackets that define the observed region were $f_1(t) = e^{(3t)} + 5e^{(-2t)}$ with bracket $[0, 1]$, $f_2(t) = \ln(t)^2 - 2 + \ln(10-t)^2 - t^{0.2}$ with bracket $[6, 9.9]$, $f_3(t) = -3t\sin(0.75t) + e^{(-2t)}$ with bracket $[0, 2\pi]$. The first and second derivatives of each objective function were using the symbolic math toolbox in MATLAB and then copied and pasted over to the main code. This was done since doing calculations using symbolic expressions made the iterative algorithms extremely slow. For each function, the starting estimate t_0 was selected such that it was within the bracket of the function and has the largest value from the objective function where $f_x(t_0) > f_x(t_1)$. With the help of the graphing calculator on Desmos, the t_0 chosen for f_1 was 1, f_2 was 9.9, and f_3 was 2π.

3.1 Quadratic Approximation

For each objective function, a matrix like the one shown on the very left-hand side of equation (2) is initialized with $t = t_0$. The other part of these systems of equations was set up by evaluating the objective function, its first and second derivative at t_0 and gathering these values into a vector such that it lines up with the respective rows in the previously generated matrix. Then this system was solved using the built-in backlash operator in MATLAB which solves linear equations in form $ax = b$. The result of this calculation is a vector whose entries correspond to the coefficients of the quadratic model. The minimizer of the outputted model is calculated with the equation described previously $\hat{t} = \frac{-a_2}{2a_1}$ where a_1 is the coefficient of the second order term of the approximation function and a_2 is the coefficient of the first order term. The accuracy of this approximation was evaluated based on the magnitude of the difference between the estimated minimizer and the actual local minimizer.

3.2 Fixed Step Size Line Search

For each objective function, the algorithm receives four parameters from the user. The starting estimate which is initialized as the previously mentioned t_0 value and the step size s which was chosen to be a hundredth of the bracket width. The two other parameters involved in the convergence criteria. Firstly is that the maximum number of iterations has been limited to 50000 and secondly the magnitude of the slope of the gradient $f_x'(t_k)$ must be equal to or less than 10^{-6} and lastly. So starting at k_0, the algorithm will run while the $k < 50000$ and $|f_x'(t_k)| \leq 10^{-6}$. To start the algorithm, initialize the current minimizer estimate as $curr\_t = t_0$, the current $f_x(t)$ as $curr\_f$, and the current $f_x'(t)$ as $curr\_g$. In each iteration, $curr\_t$ is updated to $curr\_t + (s * -curr\_g)$, then $curr\_f$ and $curr\_g$ are updated by evaluating at the new $curr\_t$, and finally k is incremented by 1. The effectiveness of the estimation on each objective function is evaluated on whether the estimate converges or not and if it does, the magnitude of the difference between the estimated minimizer and the actual local minimizer, and the number of iterations.

3.3 Line Search with Armijo Backtracking

Line search with Armijo backtracking has the same parameters mentioned previously with the change to a different s value of a tenth of the bracket width and the addition of the back-off rate β and the Armijo condition coefficient α which were both set to 0.5; note that this differs from the previous definition of $\alpha := \frac{f'(t)}{2}$, here α does not include the derivative component. The process for a line search with Armijo backtracking is virtually identical except that within iteration before the current estimate is updated, the function is evaluated at $curr\_t + s * -curr\_g$; this value will be denoted as f_est. While f_est is greater than or equal to the current estimated minimizer plus the Armijo condition, each iteration will reduce s by a factor of β and update f_est in the same way based on the new step size. After γ iterations, this s_γ value will be used as the new step size for the update of the current estimate. The rest of the algorithm carries identically to the fixed step size search. The effectiveness of the estimation on each objective function is evaluated on whether the estimate converges or not and if it does, the magnitude of the difference between the estimated minimizer and the actual local minimizer, and the number of iterations.

4 Results

Table 1: Numerical results rounded to three decimal places for one step of quadratic approximation, fixed step size search, and Armijo backtracking search.

Method	**Functions (iterations)**		
	f_1	f_2	f_3
Quadratic	0.679 (1)	9.830 (1)	6.566 (1)
Fixed	0.241 (43)	8.624 (500)	2.707 (38)
Backtracking	0.241 (13)	8.624 (42)	2.707 (6)

5 Discussion

For objective function $f_1(t)$ with bracket $[0, 1]$, starting estimate $t_0 = 1$, step size of $\frac{1}{100}$ for fixed step size search, and $\frac{1}{10}$ for Armijo backtracking search, the actual local minimizer rounded to three decimal places was $t^* = 0.241$. The quadratic approximation gave an estimated minimizer of $\hat{t} = 0.679$, the fixed step size line search did converge. It outputted an estimate of $t_F = 0.241$ after 43 iterations, and the line search with Armijo backtracking converged and outputted an estimate of $t_B = 0.241$ after 13 iterations. For objective function $f_2(t)$ with bracket $[6, 9.9]$ and starting estimate $t_0 = 9.9$, step size of $\frac{3.9}{100}$ for fixed step size search, and $\frac{3.9}{10}$ for Armijo backtracking search, the actual local minimizer rounded to three decimal places was $t^* = 8.624$. The quadratic approximation gave an estimated minimizer of $\hat{t} = 9.830$, the fixed step size line search did converge. It outputted an estimate of $t_F = 8.624$ after 500 iterations, and the line search with Armijo backtracking converged and outputted an estimate of $t_B = 8.624$ after 42 iterations. For objective function $f_3(t)$ with bracket $[0, 2\pi]$ and starting estimate $t_0 = 2\pi$, step size of $\frac{2\pi}{100}$ for fixed step size search, and $\frac{2\pi}{10}$ for Armijo backtracking search,the actual local minimizer rounded to three decimal places is $t^* = 2.706$. The quadratic approximation gave an estimated minimizer of $\hat{t} = 6.566$, the fixed step size line search did converge. It outputted an estimate of $t_F = 2.707$ after 38 iterations, and the line search with Armijo backtracking converged and outputted an estimate of $t_B = 2.707$ after 6 iterations. All outputs were rounded to three decimal places.

This implementation of quadratic approximation using only one point performed poorly compared to the line search algorithms for all three objective functions with each respective t_0. For f_1 the magnitude of the difference between $t*$ and $\hat{t}$ is $|0.241 - 0.679| = 0.438$, for f_2 the magnitude of the difference between $t*$ and $\hat{t}$ is $|8.624 - 9.830| = 1.206$, and for f_3 the magnitude of the difference between $t*$ and $\hat{t}$ is $|2.706 - 6.566| = 3.860$. Compare this with the estimated minimizers provided by the fixed step size and Armijo backtracking search which when rounded to the three decimal places are virtually identical to the actual minimizer. The estimated minimizer provided by the two line search algorithms for f_3 do differ by 0.001 but this is not a significant error and can likely be attributed to rounding errors in the calculations made by MATLAB and/or Desmos. One intuitive suggestion for improving upon this implementation of quadratic approximation would be to pick a better starting estimate, for example taking the midpoint of the bracket range. Another possible improvement would be to bootstrap this model into a quadratic model with three estimate points. This can accomplished by performing quadratic approximation the same way, then using the resultant $\hat{t}$ has another estimate point t_1 to do a second quadratic approximation using $f(t_0)$, $f(t_1)$, and the first derivative of either. This would provide a third point t_2 which then can be used to calculate a quadratic approximation using three estimate points.

For the two line search algorithms, it can be seen that Armijo backtracking significantly reduces the number of iterations needed to reach the convergence criteria. The difference is especially notable for f_2 where the number of iterations for the Armijo backtracking search is less than a tenth of the fixed step size search. Although backtracking can improve gradient descent by increasing the convergence speed, it comes with a disadvantage of extra computational cost which could be detrimental for cases where efficiency is of more importance. Additionally, Armijo backtracking includes extra hyper-parameters such as the user inputted α, β, and s values described previously which would require effort to be tuned.

For the given objective functions, it was found that the estimated minimizer given by one step of a quadratic approximation using one point, its first and second derivatives performed poorly compared to the estimates provided by the fixed step size line search and the Armijo backtracking line search in terms of the difference between the estimated minimizer and the actual minimizer. The quadratic approximation could be improved upon by providing a better starting estimate and/or bootstrapping into three estimate points. For the two line search algorithms, it was found that both algorithms converged and Armijo backtracking reduced the number of iterations until convergence by a significant amount by dynamically estimating the step size for each iteration.

References

[Arm66] Larry Armijo. "Minimization of functions having Lipschitz continuous first partial derivatives". In: *Pacific Journal of Mathematics* 16.1 (Jan. 1, 1966), pp. 1–3. DOI: 10.2140/pjm.1966.16.1. URL: https://doi.org/10.2140/pjm.1966.16.1.