Mathematical Modeling, Proof of Concept

Mathematical Modeling

Gerhard Dangelmayr and Michael Kirby

Department of Mathematics, Colorado State University, Fort Collins, Colorado 80524

E-mail address: gerhard@math.colostate.edu, kirby@math.university.edu

This project is supported in part by NSF Grant DUE-#0126650.

Contents

CHAPTER 6

Nonlinear Programming

A nonlinear programming problem has the form

$$\min f(x)$$

subject to the inequality constraints

$$g_i(x) \leq 0$$

for $i = 1, \ldots, p$ and the equality constraints

$$h_i(x) = 0$$

for $i = 1, \ldots, q$.

EXAMPLE 6.1. Given a fixed area of cardboard A construct a box of maximum volume. The nonlinear program for this is

$$\min xyz$$

subject to

$$2xy + 2xz + 2yz = A$$

EXAMPLE 6.2. Consider the problem of determining locations for two new high schools in a set of P subdivisions N_j. Let w_{1j} be the number of students going to school A and w_{2j} be the number of students going to school B from subdivision N_j. Assume that the student capacity of school A is c_1 and the capacity of school B is c_2 and that the total number of students in each subdivision is r_j. We would like to minimize the total distance traveled by all the students given that they may attend either school A or B. It is possible to construct a nonlinear program to determine the locations (a, b) and (c, d) of high schools A and B, respectively assuming the location of each subdivision N_i is model as a single point denoted (x_i, y_i).

$$\min \sum_{j=1}^{P} w_{1j} \left((a - x_j)^2 + (b - y_j)^2 \right)^{\frac{1}{2}} + w_{2j} \left((c - x_j)^2 + (d - y_j)^2 \right)^{\frac{1}{2}}$$

subject to the constraints

$$\sum_j w_{ij} \leq c_i$$

$$w_{1j} + w_{2j} = r_j$$

for $j = 1, \ldots, P$.

EXAMPLE 6.3. Neural networks have provided a new tool for approximating functions where the functional form is unknown. The approximation takes on the form

$$f(x) = \sum_j b_j \sigma(a_j x - \alpha_j) - \beta$$

and the corresponding sum of squares error term is

$$E(a_j, b_j, \alpha_j, \beta) = \sum_i \big(y_i - f(x_i)\big)^2$$

The problem of minimizing the error function is, in this instance, an unconstrained optimization problem. An efficient means for computing the gradient of E is known as the *backpropogation* algorithm.

6.1. Unconstrained Optimization in One Dimension

Here we begin by considering a significantly simplified (but nonetheless important) nonlinear programming problem, i.e., the domain and range of the function to be minimized are one-dimensional and there are no constraints. A necessary condition for a minimum of a function was developed in calculus and is simply

$$f'(x) = 0$$

Note that higher derivative tests can determine whether the function is a max or a min, or the value $f(x + \delta)$ may be compared to $f(x)$.

Note that if we let

$$g(x) = f'(x)$$

then we may convert the problem of finding a minimum or maximum of a function to the problem of finding a zero.

6.1.1. Bisection Algorithm.

Let x^* be a root, or zero, of $g(x)$, i.e., $g(x^*) = 0$. If an initial bracket $[a, b]$ is known such that $x^* \in [a, b]$, then a simple and robust approach to determining the root is to bisect this interval into two intervals $[a, c]$ and $[c, b]$ where c is the midpoint, i.e.,

$$c = \frac{a + b}{2}$$

If

$$g(a)g(c) < 0$$

then we conclude

$$x^* \in [a, c]$$

while if

$$g(b)g(c) < 0$$

then we conclude

$$x^* \in [b, c]$$

This process may now be iterated such that the size of the bracket (as well as the actual error of the estimate) is being divided by 2 every iteration.

6.1.2. Newton's Method.

Note that in the bisection method the actual value of the function $g(x)$ was only being used to determine the correct bracket for the root. Root finding is accelerated considerably by using this function information more effectively.

For example, image we were seeking the root of a function that was a straight line, i.e., $g(x) = ax + b$ and our initial guess for the root was x_0. If we extend this straight line from the point x_0 it is easy to determine where it crosses the axis, i.e.,

$$ax_1 + b = 0$$

so $x_1 = -b/a$. Of course, if the function were truly linear then no first guess would be required. So now consider the case that $g(x)$ is nonlinear but may be approximated locally about the point x_0 by a line. Then the point of intersection of this line with the x-axis is an estimate, or second guess, for the root x^*. The linear approximation comes from Taylor's theorem, i.e.,

$$g(x) = g(x_0) + g'(x_0)(x - x_0) + \frac{1}{2}g''(x_0)(x - x_0)^2 + \dots$$

So the linear approximation to $g(x)$ about the point x_0 can be written

$$l(x) = g(x_0) + g'(x_0)(x - x_0)$$

If we take x_1 to be the root of the linear approximation we have

$$l(x_1) = 0 = g(x_0) + g'(x_0)(x_1 - x_0)$$

Solving for x_1 gives

$$x_1 = x_0 - \frac{g(x_0)}{g'(x_0)}$$

or at the nth iteration

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}$$

The iteration above is for determining a zero of a function $g(x)$. To determine a maximum or minimum value of a function f we employ condition that $f'(x) = 0$. Now the iteration is modified as as

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

6.2. Unconstrained Optimization in Higher Dimensions

Now we consider the problem of minimizing (or maximizing) a scalar function of many variables, i.e., defined on a vector field. We consider the extension of Newton's method presented in the previous section as well as a classical approach known as steepest descent.

6.2.1. Taylor Series in Higher Dimensions. Before we extend the search for extrema to higher dimensions we present Taylor series for functions of more than one domain variable. To begin, the Taylor series for a function of two variables is given by

$$\begin{aligned} g(x,y) =& g(x^{(0)}, y^{(0)}) + \frac{\partial g}{\partial x}(x - x^{(0)}) + \frac{\partial g}{\partial y}(y - y^{(0)}) \\ &+ \frac{\partial^2 g}{\partial x^2}\frac{(x - x^{(0)})^2}{2} + \frac{\partial^2 g}{\partial y^2}\frac{(y - y^{(0)})^2}{2} + \frac{\partial^2 g}{\partial x \partial y}(x - x^{(0)})(y - y^{(0)}) \\ &+ \text{higher order terms} \end{aligned}$$

In n variables $x = (x_1, \dots, x_n)$ the Taylor series expansion becomes

$$g(x) = g(x^{(0)}) + \nabla g(x^{(0)})(x - x^{(0)}) + \frac{1}{2}(x - x^{(0)})^T Hg(x^{(0)})(x - x^{(0)}) + \cdots$$

where the Hessian matrix is defined as

$$\left(Hg(x^{(0)})\right)_{ij} = \left.\frac{\partial^2 g}{\partial x_i \partial x_j}\right|_{x = x^{(0)}}$$

and the gradient is written as a row vector, i.e.,

$$\big(\nabla g(x^{(0)})\big)_i = \frac{\partial g}{\partial x_i}\bigg|_{x=x^{(0)}}$$

6.2.2. Roots of a Nonlinear System. We saw that Newton's method could be used to develop an iteration for determining the zeros of a scalar function. We can extend those ideas for determining roots of the nonlinear system

$$\begin{aligned} g_1(x_1,\ldots,x_n) &= 0 \\ g_2(x_1,\ldots,x_n) &= 0 \\ &\vdots \\ g_n(x_1,\ldots,x_n) &= 0 \end{aligned}$$

or, more compactly,

$$g(x) = 0.$$

Now we apply Taylor's theorem to each component $g_i(x_1,\ldots,x_n)$ individually, i.e., retaining only the first order terms we have the linear approximation to g_i about the point $x^{(0)}$ as

$$l_i(x) = g_i(x^{(0)}) + \nabla g_i(x^{(0)})(x - x^{(0)})$$

for $i = 1,\ldots,n$. We can write these components together as a vector equation

$$l(x) = g(x^{(0)}) + \nabla g(x^{(0)})(x - x^{(0)})$$

where now

$$\big(\nabla g(x^{(0)})\big) = \frac{\partial g_i}{\partial x_j}\bigg|_{x=x^{(0)}}$$

As in the scalar case we base our iteration on the assumption that

$$l(x^{(k+1)}) = 0$$

Hence,

$$g(x^{(k)}) + \nabla g(x^{(k)})(x^{(k+1)} - x^{(k)}) = 0$$

and given $x^{(k)}$ we may determine the next iterate $x^{(k+1)}$ by solving an $n \times n$ system of equations.

6.2.3. Newton's Method. In this chapter we are interested in minimizing functions of several variables. Analogously with the scalar variable case we may modify the above root finding method to determine maximum (or minima) of a function $f(x_1,\ldots,x_n)$. To compute an extreme point we require that $\nabla f = 0$ hence we set

$$g(x) = \big(\frac{\partial f}{\partial x_1},\ldots,\frac{\partial f}{\partial x_n}\big)^T$$

Substituting

$$g_i(x) = \frac{\partial f}{\partial x_i}$$

into

$$g(x^{(k)}) + \nabla g(x^{(k)})(x^{(k+1)} - x^{(k)}) = 0$$

produces

$$\nabla f(x^{(k)}) + Hf(x^{(k)})(x^{(k+1)} - x^{(k)}) = 0$$

where

$$\left(Hf(x^{(k)})\right)_{ij} = \frac{\partial g_i}{\partial x_j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Again we have a linear system for $x^{(k+1)}$.

6.2.4. Steepest Descent. Another form for Taylor's formula in n-variables is given by

$$f(x+th) = f(x) + t\nabla f(x)h + \frac{t^2}{2}h^T Hf(x)h$$

where again $(\nabla f(x))_i = \partial f/\partial x_i$ and $(Hf(x))_{ij} = \partial^2 f/\partial x_i \partial x_j$. Now t is a scalar and $x+th$ is a ray emanating from the point x in the direction h. We can compute the derivative of the function f w.r.t. t as

$$\frac{df}{dt}(x+th) = \nabla f(x)h + th^T Hf(x)f$$

Evaluating the derivative at the point $t = 0$ gives

$$\frac{df}{dt}(x) = \nabla f(x)h$$

This quantity, known as the directional derivative, indicates how the function is changing at the point x in the direction h. Recall from calculus that the direction of maximum increase (decrease) of a function is in the direction of the gradient (negative gradient). This is readily seen from the formula for the directional derivative using the identity

$$\nabla f(x)h = \|\nabla f(x)\| \|h\| \cos(\theta)$$

We can assume without loss of generality that h is of unit length, i.e., $\|h\| = 1$. So the quantity on the right is a maximum when the vectors h and $\nabla f(x)$ point in the same direction so $\theta = 0$.

This observation may be used to develop an algorithm for unconstrained function minimization. With an appropriate choice of the scalar step-size α, the iterations

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)}) \tag{6.1}$$

will converge (possibly slowly) to a minimum of the function $f(x)$.

6.3. Constrained Optimization and Lagrange Multipliers

Consider the constrained minimization problem

$$\min f(x)$$

subject to

$$c_i(x) = 0$$

$i = i, \ldots, p$. It can be shown that a necessary condition for a solution to this problem is provided by solving

$$\nabla f = \lambda_1 \nabla c_1 + \cdots + \lambda_p \nabla c_p$$

where the λ_i are referred to as Lagrange multipliers. Consider the case of f, c being functions of two variables and consider their level curves. A little thought reveals that the minimum value of f on the constraint c is given when the gradient of each function is parallel.

We demonstrate a solution via this procedure by recalling our earlier example.

EXAMPLE 6.4. Given a fixed area of cardboard A construct a box of maximum volume. The nonlinear program for this is

$$\min xyz$$

subject to

$$2xy + 2xz + 2yz = A$$

Now $f(x, y, z) = xyz$ and $c(x, y, z) = 2xy + 2yz + 2xz - A$. Substituting these functions into our condition gives

$$\nabla f = \lambda \nabla c$$

produces the system of equations

$$\begin{aligned} yz - \lambda(2y + 2z) &= 0 \\ xz - \lambda(2x + 2z) &= 0 \\ xy - \lambda(2y + 2x) &= 0 \end{aligned}$$

The solution to this system may be computed analytically to be

$$x = y = z = \sqrt{\frac{A}{6}}$$

In this special case the nonlinear system could be solved by hand. Typically this is not the case and one must resort to numerical techniques such as Newton's method to solve the resulting $(n + m) \times (n + m)$ system

$$g(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_m) = 0.$$

PROBLEM 6.1. Extend Example 6.2 for a collection of S schools.

PROBLEM 6.2. Show how Newton's method for root finding can be used to calculate $\sqrt{3}$. Use Matlab to compute the sequence that converges to this value. What is the effect of changing the initial condition?

PROBLEM 6.3. Minimize the objective function

$$f(x_1, x_2) = 7x_1^2 + 2x_1x_2 + x_2^2$$

using 50 iterations of

a) Newton's method
b) Steepest Descent

Plot the value of the iterates for each method on the same graph. You may experiment with the value of α in Equation (6.1). Hint start small.

PROBLEM 6.4. Assume a farmer has L feet of fencing for a rectangular area with lengths x and y. Determine these lengths such that the enclosed area is a maximum.