Optimization Techniques

UNCONSTRAINED NLP EXAMPLE

Consider the problem:

Minimize $f(x_1, x_2, x_3) = (x_1)^2 + x_1(1 - x_2) + (x_2)^2 - x_2x_3 + (x_3)^2 + x_3$

First, we find the gradient with respect to x_i:

$$\nabla f = \begin{bmatrix} 2x_1 + 1 - x_2 \\ -x_1 + 2x_2 - x_3 \\ -x_2 + 2x_3 + 1 \end{bmatrix}$$

Optimization Techniques

UNCONSTRAINED NLP EXAMPLE

Next, we set the gradient equal to zero:

$$\nabla f = 0 \qquad \Rightarrow \qquad \begin{bmatrix} 2x_1 + 1 - x_2 \\ -x_1 + 2_2 - x_3 \\ -x_2 + 2x_3 + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

So, we have a system of 3 equations and 3 unknowns. When we solve, we get:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

Optimization Techniques

UNCONSTRAINED NLP EXAMPLE

So we have only one candidate point to check.

Find the Hessian:

$$\nabla^2 f = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

Optimization Techniques

UNCONSTRAINED NLP EXAMPLE

The eigenvalues of this matrix are:

$$\lambda_1 = 3.414 \qquad \lambda_2 = 0.586 \qquad \lambda_3 = 2$$

All of the eigenvalues are > 0, so the Hessian is positive definite.

So, the point $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$ is a minimum

Optimization Techniques

UNCONSTRAINED NLP EXAMPLE

Unlike in Linear Programming, unless we know the shape of the function being minimized or can determine whether it is convex, we cannot tell whether this point is the global minimum or if there are function values smaller than it.

Optimization Techniques

9.2: Taylor Series

Brook Taylor was an accomplished musician and painter. He did research in a variety of areas, but is most famous for his development of ideas regarding infinite series.

Brook Taylor
1685 - 1731

Greg Kelly, Hanford High School, Richland, Washington

Optimization Techniques

Suppose we wanted to find a fourth degree polynomial of the form:

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$

that approximates the behavior of $f(x) = \ln(x+1)$ at $x = 0$

If we make $P(0) = f(0)$ and the first, second, third and fourth derivatives the same, then we would have a pretty good approximation.

→

Optimization

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \qquad f(x) = \ln(x+1)$$

$$f(x) = \ln(x+1) \qquad P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$

$$f(0) = \ln(1) = 0 \qquad P(0) = a_0 \rightarrow \boxed{a_0 = 0}$$

$$f'(x) = \frac{1}{1+x} \qquad P'(x) = a_1 + 2a_2 x + 3a_3 x^2 + 4a_4 x^3$$

$$f'(0) = \frac{1}{1} = 1 \qquad P'(0) = a_1 \rightarrow \boxed{a_1 = 1}$$

$$f''(x) = -\frac{1}{(1+x)^2} \qquad P''(x) = 2a_2 + 6a_3 x + 12a_4 x^2$$

$$f''(0) = -\frac{1}{1} = -1 \qquad P''(0) = 2a_2 \rightarrow \boxed{a_2 = -\frac{1}{2}}$$

→

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \qquad f(x) = \ln(x+1)$$

$$f''(x) = -\frac{1}{(1+x)^2} \qquad P''(x) = 2a_2 + 6a_3 x + 12a_4 x^2$$

$$f''(0) = -\frac{1}{1} = -1 \qquad P''(0) = 2a_2 \rightarrow \boxed{a_2 = -\frac{1}{2}}$$

$$f'''(x) = 2 \cdot \frac{1}{(1+x)^3} \qquad P'''(x) = 6a_3 + 24a_4 x$$

$$f'''(0) = 2 \qquad P'''(0) = 6a_3 \rightarrow \boxed{a_3 = \frac{2}{6}}$$

$$f^{(4)}(x) = -6\frac{1}{(1+x)^4} \qquad P^{(4)}(x) = 24a_4$$

$$f^{(4)}(0) = -6 \qquad P^{(4)}(0) = 24a_4 \rightarrow \boxed{a_4 = -\frac{6}{24}}$$

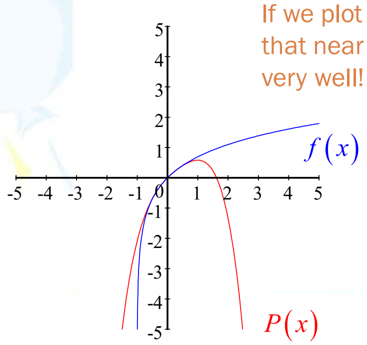
→

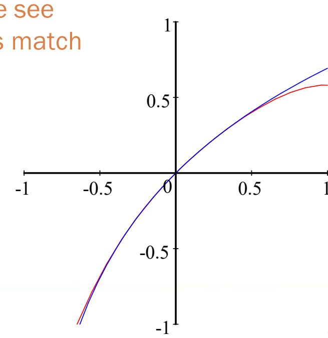
$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \qquad f(x) = \ln(x+1)$$

$$P(x) = 0 + 1x - \frac{1}{2}x^2 + \frac{2}{6}x^3 - \frac{6}{24}x^4$$

$$P(x) = 0 + x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} \qquad f(x) = \ln(x+1)$$

If we plot both functions, we see that near zero the functions match very well!





→

Optimization Techniques

Our polynomial: $$0+1x-\frac{1}{2}x^2+\frac{2}{6}x^3-\frac{6}{24}x^4$$

has the form: $$f(0)+f'(0)x+\frac{f''(0)}{2}x^2+\frac{f'''(0)}{6}x^3+\frac{f^{(4)}(0)}{24}x^4$$

or: $$\frac{f(0)}{0!}+\frac{f'(0)}{1!}x+\frac{f''(0)}{2!}x^2+\frac{f'''(0)}{3!}x^3+\frac{f^{(4)}(0)}{4!}x^4$$

This pattern occurs no matter what the original function was!

→

Optimization

Maclaurin Series:

(generated by f at $x=0$)

$$P(x)=f(0)+f'(0)x+\frac{f''(0)}{2!}x^2+\frac{f'''(0)}{3!}x^3+\cdots$$

If we want to center the series (and it's graph) at some point other than zero, we get the Taylor Series:

Taylor Series:

(generated by f at $x=a$)

$$P(x)=f(a)+f'(a)(x-a)+\frac{f''(a)}{2!}(x-a)^2+\frac{f'''(a)}{3!}(x-a)^3+\cdots$$

→

Optimization Techniques

example: $y=\cos x$

$$f(x)=\cos x \qquad f(0)=1 \qquad f'''(x)=\sin x \qquad f'''(0)=0$$

$$f'(x)=-\sin x \qquad f'(0)=0 \qquad f^{(4)}(x)=\cos x \qquad f^{(4)}(0)=1$$

$$f''(x)=-\cos x \qquad f''(0)=-1$$

$$P(x)=1+0x-\frac{1x^2}{2!}+\frac{0x^3}{3!}+\frac{1x^4}{4!}+\frac{0x^5}{5!}-\frac{1x^6}{6!}+\cdots$$

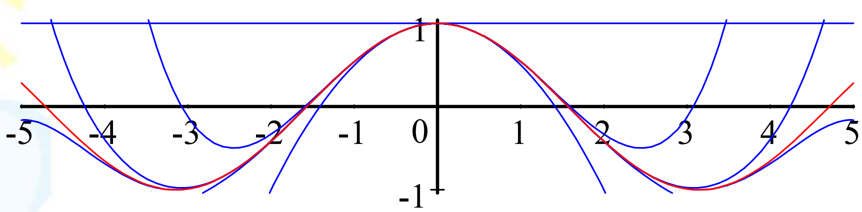
$$\boxed{P(x)=1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\frac{x^8}{8!}-\frac{x^{10}}{10!}\cdots}$$

→

Optimization Techniques

$y=\cos x$

$$P(x)=1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\frac{x^8}{8!}-\frac{x^{10}}{10!}\cdots$$



The more terms we add, the better our approximation.

Hint: On the TI-89, the factorial symbol is:

→

Optimization Techniques

example: $y = \cos(2x)$

Rather than start from scratch, we can use the function that we already know:

$$P(x) = 1 - \frac{(2x)^2}{2!} + \frac{(2x)^4}{4!} - \frac{(2x)^6}{6!} + \frac{(2x)^8}{8!} - \frac{(2x)^{10}}{10!} \cdots$$

Optimization Techniques

example: $y = \cos(x)$ at $x = \frac{\pi}{2}$

$$f(x) = \cos x \quad f\left(\frac{\pi}{2}\right) = 0$$

$$f'(x) = -\sin x \quad f'\left(\frac{\pi}{2}\right) = -1$$

$$f''(x) = -\cos x \quad f''\left(\frac{\pi}{2}\right) = 0$$

$$f'''(x) = \sin x \quad f'''\left(\frac{\pi}{2}\right) = 1$$

$$f^{(4)}(x) = \cos x \quad f^{(4)}\left(\frac{\pi}{2}\right) = 0$$

$$P(x) = 0 - 1\left(x - \frac{\pi}{2}\right) + \frac{0}{2!}\left(x - \frac{\pi}{2}\right)^2 + \frac{1}{3!}\left(x - \frac{\pi}{2}\right)^3 + \cdots$$

$$\boxed{P(x) = -\left(x - \frac{\pi}{2}\right) + \frac{\left(x - \frac{\pi}{2}\right)^3}{3!} - \frac{\left(x - \frac{\pi}{2}\right)^5}{5!} + \cdots}$$

Optimization Techniques

There are some Maclaurin series that occur often enough that they should be memorized. They are on your formula sheet.

→

Optimization

When referring to Taylor polynomials, we can talk about **number of terms**, **order** or **degree**.

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!}$$

This is a polynomial in **3 terms**.

It is a **4th order** Taylor polynomial, because it was found using the 4th derivative.

It is also a **4th degree** polynomial, because x is raised to the 4th power.

The **3rd order** polynomial for $\cos x$ is $1 - \frac{x^2}{2!}$, but it is **degree 2**.

A recent AP exam required the student to know the difference between *order* and *degree*.

→

Optimization Techniques

The TI-89 finds Taylor Polynomials:

taylor (expression, variable, order, [point])

$\underbrace{\text{F3} \quad 9}$

F3 9

taylor $\left(\cos(x), x, 6\right)$ $\quad \dfrac{-x^6}{720}+\dfrac{x^4}{24}-\dfrac{x^2}{2}+1$

taylor $\left(\cos(2x), x, 6\right)$ $\quad \dfrac{-4x^6}{45}+\dfrac{2x^4}{3}-2x^2+1$

taylor $\left(\cos(x), x, 5, \pi/2\right)$ $\quad \dfrac{-(2x-\pi)^5}{3840}+\dfrac{(2x-\pi)^3}{48}-\dfrac{2x-\pi}{2}$

π

Optimization Techniques

Summary

- A first order approximation of $f(x)$ **around point** $\bar{x}$ is given by
- $f_1(x) = f(x') + f'(x') * (x - x')$
- *Taylor Approximation around a vector x*
- $f_1(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^T(\boldsymbol{x} - \boldsymbol{x}^k)$
- $f1(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \sum_{j=1}^{n} \frac{\partial f(x^k)}{\partial x_j}(x_j - x_j^k)$
- Determine first order Taylor approximation of the function
  - $f(x_1, x_2) = x_1^4 + x_1^2 + 2x_2^2 - 2x_1x_2$ around a point (x_1, x_2)=(1,1)

Optimization Techniques

Summary

- A second order approximation of $f(x)$ **around point** $\bar{x}$ for single variable is given by
- $f_2(x) = f(x') + f'(x')(x - x') + \frac{1}{2} f''(x')(x - x')^2$
- *A second order Taylor Approximation around a vector x*
- $f_2(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^T (\boldsymbol{x} - \boldsymbol{x}^k) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}^k)^T H(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k)$
- $f1(\boldsymbol{x}) = f(\boldsymbol{x}^k) + \sum_{j=1}^{n} \frac{\partial f(\boldsymbol{x}^k)}{\partial x_j} * (x_j - x_j^k) + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 f(\boldsymbol{x}^k)}{\partial x_i \partial x_j} (x_i - x_i^k)(x_j - x_j^k)$

$f(x_1, x_2) = x_1^4 + x_1^2 + 2x_2^2 - 2x_1x_2$ around a point (x_1, x_2)=(1,1)

Optimization Techniques

METHOD OF SOLUTION

- In the previous example, when we set the gradient equal to zero, we had a system of 3 linear equations & 3 unknowns.
- For other problems, these equations could be nonlinear.
- Thus, the problem can become trying to solve a system of nonlinear equations, which can be very difficult.

Optimization Techniques

METHOD OF SOLUTION

- To avoid this difficulty, NLP problems are usually solved numerically.
- We will now look at examples of numerical methods used to find the optimum point for single-variable NLP problems. These and other methods may be found in any numerical methods reference.

Optimization Techniques

NEWTON'S METHOD

When solving the equation $f'(x) = 0$ to find a minimum or maximum, one can use the iteration step:

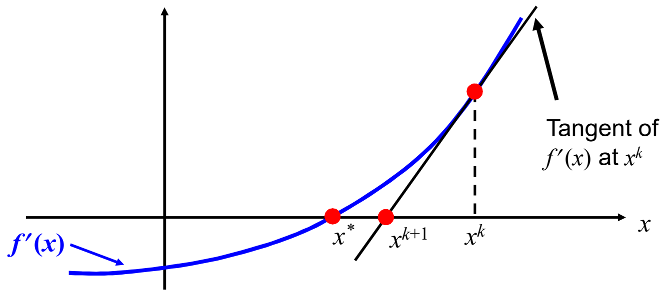
$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}$$

where k is the current iteration.

Iteration is continued until $|x^{k+1} - x^k| < \varepsilon$ where ε is some specified tolerance.

Optimization Techniques

NEWTON'S METHOD DIAGRAM



Newton's Method approximates $f'(x)$ as a straight line at x^k and obtains a new point (x^{k+1}), which is used to approximate the function at the next iteration. This is carried on until the new point is sufficiently close to x^*.

Optimization Techniques

NEWTON'S METHOD COMMENTS

- One must ensure that $f(x^{k+1}) < f(x^k)$ for finding a minimum and $f(x^{k+1}) > f(x^k)$ for finding a maximum.
- Disadvantages:
  - The initial guess is very important – if it is not close enough to the solution, the method may not converge
  - Both the first and second derivatives must be calculated

Optimization Techniques

REGULA-FALSI METHOD

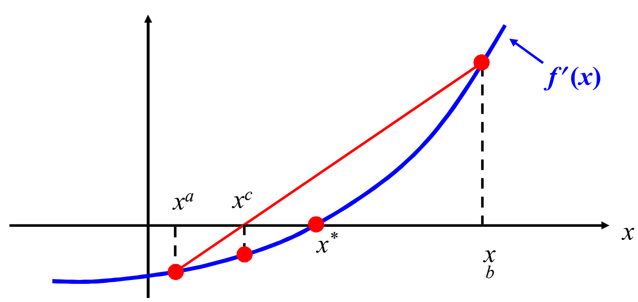
This method requires two points, x^a & x^b that bracket the solution to the equation $f'(x) = 0$.

$$x^c = x^b - \frac{f'(x^b)\cdot(x^b - x^a)}{f'(x^b) - f'(x^a)}$$

where x^c will be between x^a & x^b. The next interval will be x^c and either x^a or x^b, whichever has the sign opposite of x^c.

Optimization Techniques

REGULA-FALSI DIAGRAM



The Regula-Falsi method approximates the function $f'(x)$ as a straight line and interpolates to find the root.

Optimization Techniques

REGULA-FALSI COMMENTS

- This method requires initial knowledge of two points bounding the solution
- However, it does not require the calculation of the second derivative
- The Regula-Falsi Method requires slightly more iterations to converge than the Newton's Method

Optimization Techniques

MULTIVARIABLE OPTIMIZATION

- Now we will consider unconstrained multivariable optimization
- Nearly all multivariable optimization methods do the following:

1. Choose a search direction $\mathbf{d}^k$
2. Minimize along that direction to find a new point:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k$$

where k is the current iteration number and α^k is a positive scalar called the step size.

Optimization Techniques

THE STEP SIZE

- The step size, α^k, is calculated in the following way:
- We want to minimize the function $f(\mathbf{x}^{k+1}) = f(\mathbf{x}^k + \alpha^k \mathbf{d}^k)$ where the only variable is α^k because $\mathbf{x}^k$ & $\mathbf{d}^k$ are known.
- We set $\frac{\mathrm{d}f\left(\mathbf{x}^k + \alpha^k \mathbf{d}^k\right)}{\mathrm{d}\alpha^k} = 0$ and solve for α^k using a single-variable solution method such as the ones shown previously.

Optimization Techniques

STEEPEST DESCENT METHOD

- This method is very simple – it uses the gradient (for maximization) or the negative gradient (for minimization) as the search direction:

$$\mathbf{d}^k = \begin{Bmatrix} + \\ - \end{Bmatrix} \nabla f(\mathbf{x}^k) \text{ for } \begin{Bmatrix} \max \\ \min \end{Bmatrix}$$

So, $$\mathbf{x}^{k+1} = \mathbf{x}^k \begin{Bmatrix} + \\ - \end{Bmatrix} \alpha^k \nabla f(\mathbf{x}^k)$$

Optimization Techniques

STEEPEST DESCENT METHOD

- Because the gradient is the rate of change of the function at that point, using the gradient (or negative gradient) as the search direction helps reduce the number of iterations needed