Nonlinear Programming I

DNSC 6212: Optimization Methods and Applications

Fall 2017

Overview

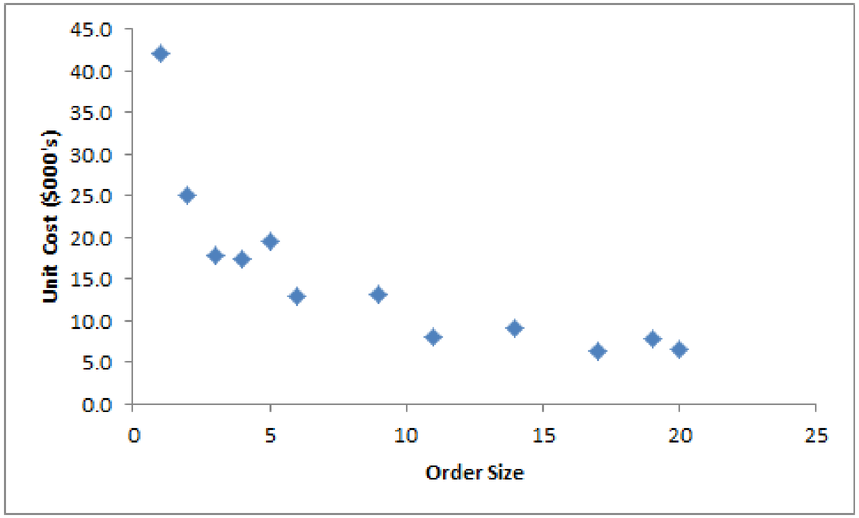
- ***Nonlinear programs*** (***NLPs***) are mathematical programs with nonlinear terms in the objective function and/or one or more of the constraints.
- An ***unconstrained NLP*** consists of simply a nonlinear objective function, while a ***constrained NLP*** additionally has constraints.
- The objective function may be ***differentiable*** allowing the usage calculus-based methods; in other cases, the objective function may be ***non-differentiable***.
- Note that an unconstrained LP is always unbounded (except in the trivial case when the objective is a constant).
- On the other hand, unconstrained NLPs can have finite optimal solutions.

Unconstrained NLP Models

Curve Fitting

- Suppose that for some company, the per unit cost for fulfilling product orders, depend on the number of units in an order.
- The table and plot on the right show the number of units and the corresponding unit cost for the last 12 orders.
- We are interested in fitting a curve to use it to estimate unit costs for future orders.

Order	Number	Unit Cost (000's)
1	19	7.9
2	2	25.0
3	9	13.1
4	4	17.4
5	5	19.5
6	6	13.0
7	3	17.8
8	11	8.0
9	14	9.2
10	17	6.3
11	1	42.0
12	20	6.6



Unconstrained NLP Models

Curve Fitting – Cont'd

- Let,

  $i \triangleq$ observation (order) number

  $m \triangleq$ number of observations (orders)

  $p_i \triangleq$ number of units for order i

  $q_i \triangleq$ unit cost for order i

- We would like to determine the ***regression*** function $r(p)$ that best explains the observations.

- Given the plot of the observations, one possibility is to fit a ***nonlinear regression*** of the form $r(p) \triangleq x_1 p^{x_2}$:
  - For a given number of units per order, p, the function returns the unit cost.
  - x_1 and x_2 are unknown parameters that need to be determined.

Unconstrained NLP Models

Curve Fitting – Cont'd

- The ***residual***, or error, associated with each of the observations *i* is:

$q_i - x_1 p_i^{x_2}$ Difference between observed unit cost and the one predicted by function.

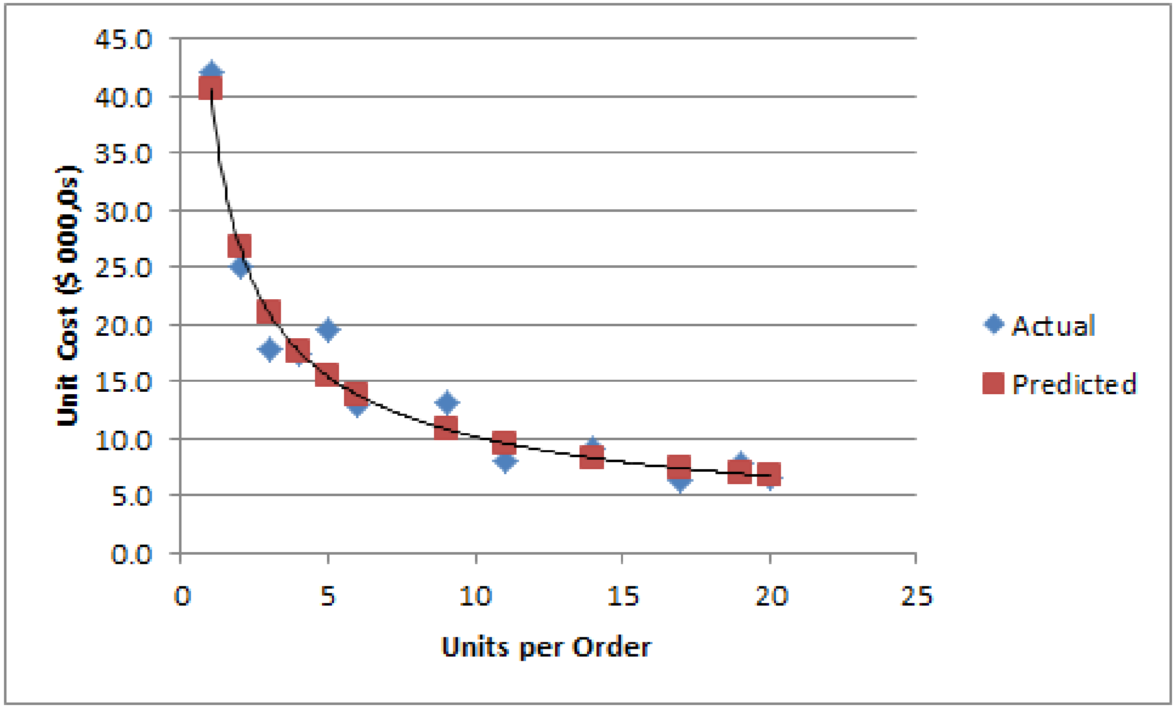
- To get the best fit, some measure of the residuals needs to be minimized.
- Simply minimizing the sum of the residuals is *not* a good idea, as positive and negative errors would cancel out.
- The most common approach is to minimize the sum of the ***squares of the residuals***:

$$\min\ f\left(x_1, x_2\right) = \sum_{i=1}^{m} \left(q_i - x_1 p_i^{x_2}\right)^2$$

Unconstrained NLP Models

Curve Fitting – Cont’d

- The optimal values for x_1 and x_2 provide the parameters for this ***least squares nonlinear*** regression (see “***Nonlinear Regression.xlsx***”): $x_1 = 40.6883$ and $x_2 = -0.6204$.
- Overlaying the fitted function over the historical data gives:

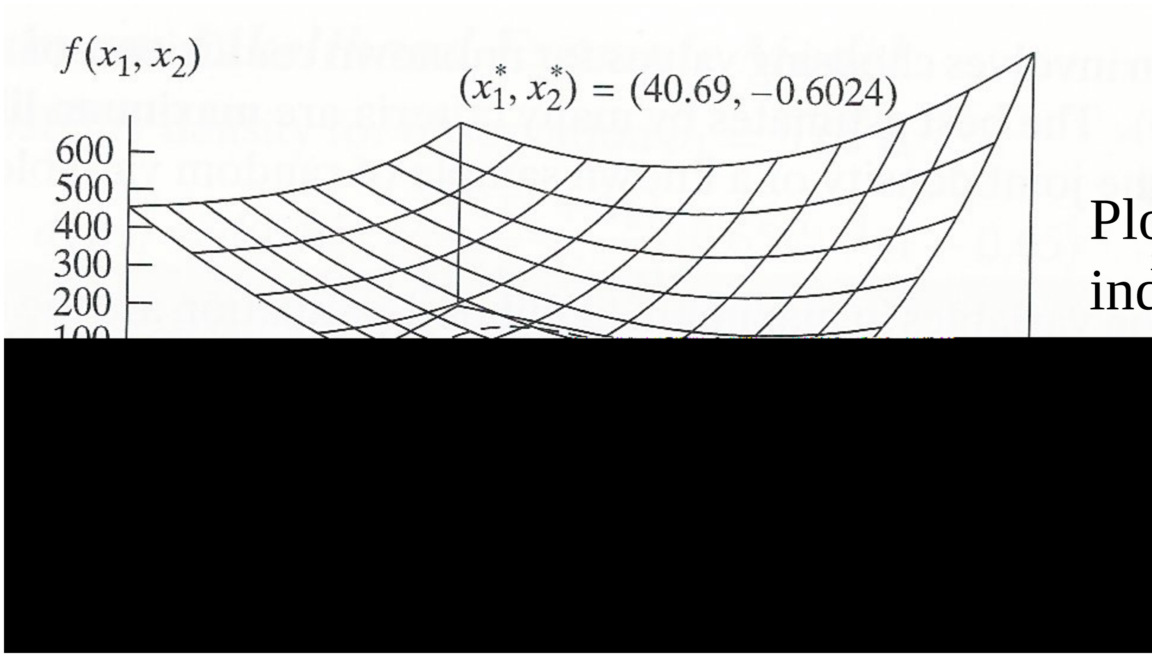


Unconstrained NLP Models

Curve Fitting – Cont’d

- The optimized objective function value is referred to as the ***root mean squared error*** (***RMSE***) and has a value 43.1

- Three-dimensional plot of the objective function:



[redacted]

Plot of objective function indicates that there is a [redacted]que optimal solution.

Unconstrained NLP Models

Maximum Likelihood Estimators

Overview

- This class of unconstrained models occurs in the context of fitting continuous probability distributions to observed data.
- A sample consists of *m* observations sampled from some hypothesized underlying distribution whose parameters are to be estimated.
- The sample can be thought of as consisting of instances of independent and identically distributed random variables P_1, P_2, …, P_m, having the same distribution $d(p)$.
- Assuming that the sample's observations are *independent*, the ***joint probability distribution function*** is of the form:

$$d\left(p_1, p_2, \cdots, p_m\right) = d\left(p_1\right) d\left(p_2\right) \cdots d\left(p_m\right).$$

Unconstrained NLP Models

Maximum Likelihood Estimators

Overview

- ***Maximum likelihood (MLE)*** estimates for the fitted distribution are ones that maximize the chance that the observations came from the distribution.

- Given the set of observations in the sample $p'_{1,} p'_{2,} \cdots, p'_m$, an unconstrained NLP is solved for the fitted function's parameters:
  - The decision variables are the parameters being estimated.
  - The objective function maximizes the likelihood that the observations came from the distribution:

    $\max d\left(p'_1\right) d\left(p'_2\right) \cdots d\left(p'_m\right)$

Unconstrained NLP Models

Maximum Likelihood Estimators

Example 1

- Observed inter-arrival times for 911 emergency calls for a local police station are 80, 10, 14, 26, 40, and 22 minutes.
- We are interested in determining the exponential probability density function distribution that best fits the observations.
- The unconstrained NLP is:

$$\max\left(\alpha e^{-\alpha t_1'}\right)\left(\alpha e^{-\alpha t_2'}\right)\left(\alpha e^{-\alpha t_3'}\right)\left(\alpha e^{-\alpha t_4'}\right)\left(\alpha e^{-\alpha t_5'}\right)\left(\alpha e^{-\alpha t_6'}\right)$$

$$= \max \alpha^6 e^{-\alpha\left(t_1'+t_2'+t_3'+t_4'+t_5'+t_6'\right)}$$

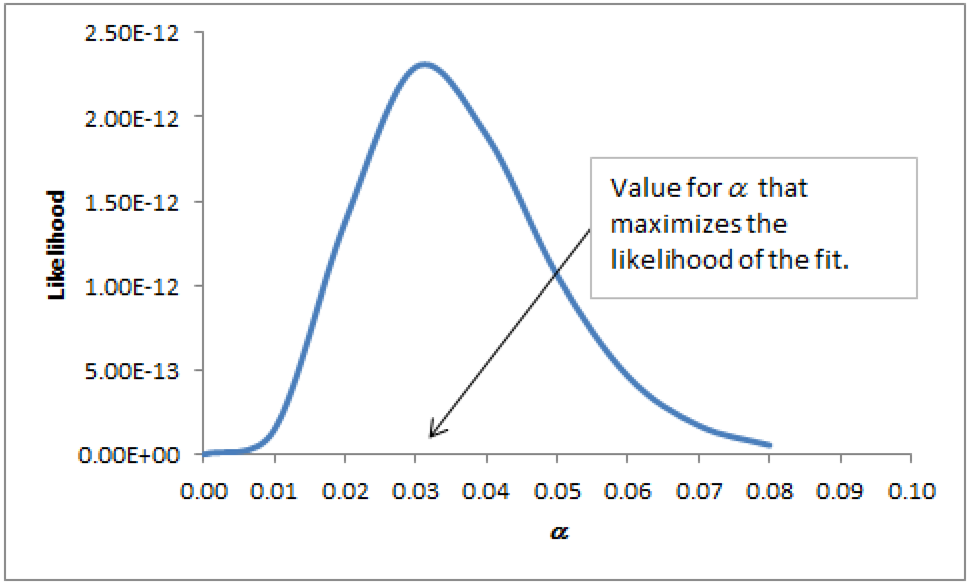
$$= \max \alpha^6 e^{-192\alpha}$$

Unconstrained NLP Models

Maximum Likelihood Estimators

Example 1

- Plotting the objective function versus α, the optimal value for α can be estimated:



- Or, the NLP can be directly solved to determine the optimal value $\alpha = 0.031$. (See "***MLE.xlsx***")

Unconstrained NLP Models

Maximum Likelihood Estimators

Example 2

- The PERT distribution is often used to estimate durations of tasks for projects.
- PERT is a Beta distribution, whereby the random variable p is interpreted as the *fraction* that an activity duration assumes out of an allowed maximum:

$$d(p) \triangleq \frac{\Gamma(x_1 + x_2)}{\Gamma(x_1)\Gamma(x_2)} (p)^{x_1 - 1} (1 - p)^{x_2 - 1}$$

where,

- x_1 and x_2 are parameters that control the shape of the distribution, and
- $\Gamma(x)$ is the standard gamma function that is equal to the area under the curve of $\gamma(x) = h^{x-1} e^{-h}$ over $0 \leq h \leq +\infty$ (i.e. $\int_0^\infty h^{x-1} e^{-h} dh$), and which has no closed form.

Maximum Likelihood Estimators

Example 2 – Cont'd

- Suppose that the following data was accumulated for the last 10 times that a project activity was undertaken, expressed as fractions in terms of the maximum allowed duration:

  0.65 0.57 0.52 0.72 0.74 0.30 0.79 0.30 0.79 0.89 0.92 0.42

- The beta probability density for the first observation:

$$d(0.65) \triangleq \frac{\Gamma(x_1+x_2)}{\Gamma(x_1)\Gamma(x_2)}(0.65)^{x_1-1}(1-0.65)^{x_2-1}$$

- The ***joint probability density function*** for the first and second observation:

$$d(0.65)d(0.57) \triangleq \left[\frac{\Gamma(x_1+x_2)}{\Gamma(x_1)\Gamma(x_2)}(0.65)^{x_1-1}(1-0.65)^{x_2-1}\right] \cdot \left[\frac{\Gamma(x_1+x_2)}{\Gamma(x_1)\Gamma(x_2)}(0.57)^{x_1-1}(1-0.57)^{x_2-1}\right]$$

Unconstrained NLP Models

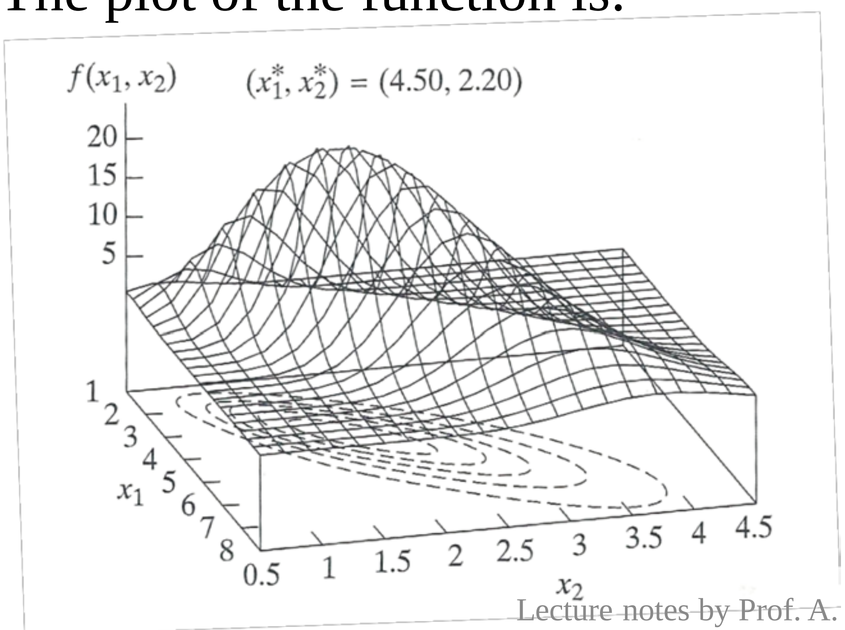
Maximum Likelihood Estimators

Example 2 – Cont'd

- Continuing in the same the NLP that maximizes the likelihood of a sample with m observations is:

$$\max\ f(x_1, x_2) \triangleq \prod_{i=1}^{m} \left[\frac{\Gamma(x_1 + x_2)}{\Gamma(x_1)\Gamma(x_2)} (p_i)^{x_1 - 1} (1 - p_i)^{x_2 - 1} \right]$$

- The plot of the function is:



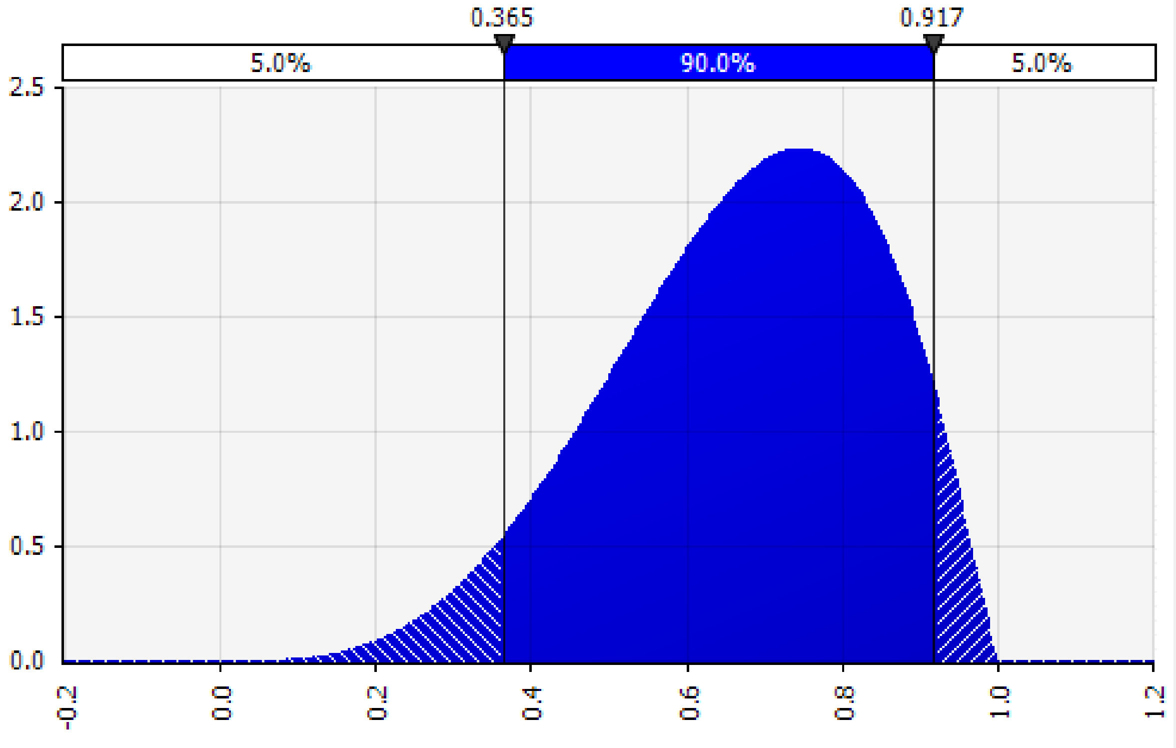
The global optimal solution is $x_1^* \approx 4.50,$ and $x_2^* \approx 2.20.$

Unconstrained NLP Models

Maximum Likelihood Estimators

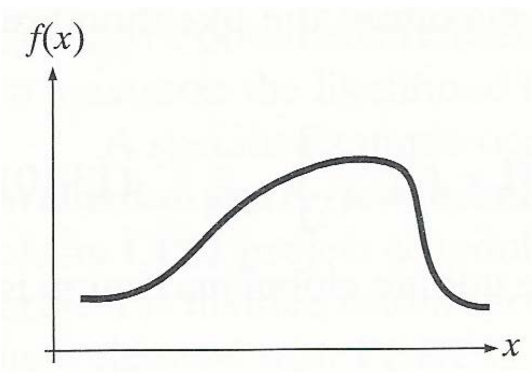
Example 2 – Cont'd

The PERT distribution fitted with the estimated parameters:

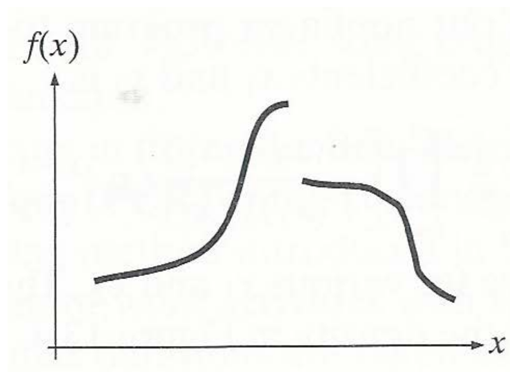


Smooth and Nonsmooth Functions

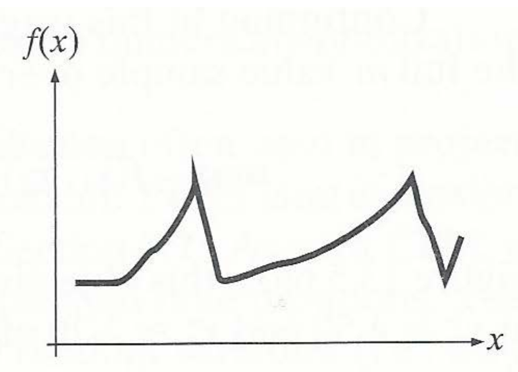
- A function is ***smooth*** if it is continuous *and* differentiable.
- A function is ***nonsmooth*** if it is discontinuous *or* nondifferentiable.
- Nonlinear programs over smooth functions are usually more tractable than those over nonsmooth functions.



Smooth



Discontinuous



Nondifferentiable

Usable Derivatives

- The fact that the function is smooth does not always mean that usable derivatives can be easily obtained to aid in search algorithms.
- For the nonlinear regression example, the NLP is:

$$\min\ f\left(x_1, x_2\right) = \sum_{i=1}^{m}\left(q_i - x_1 p_i^{x_2}\right)^2$$

The partial derivative of the objective function with respect to x_1 is:

$$\frac{\partial f}{\partial x_1} = -2\sum_{i=1}^{12}\left(q_i - x_1 p_i^{x_2}\right) p_i^{x_2}$$

The partial derivative of the objective function with respect to x_2 is:

$$\frac{\partial f}{\partial x_2} = -2\sum_{i=1}^{12}\left(q_i - x_1 p_i^{x_2}\right)\left(x_1 p_i^{x_2}\right)\ln\left(p_i\right)$$

- These partial derivative can be used to produce an efficient search.

Usable Derivatives – Cont'd

- Similarly, for the 1st MLE example, the objective function is $\max \alpha^6 e^{-192\alpha}$.
- The derivative of the objective function with respect to α is:

$$\frac{d\left(\alpha^6 e^{-192\alpha}\right)}{d\alpha} = -192\alpha^6 e^{-192\alpha} + 6\alpha^5 e^{-192\alpha}$$

- For the 2nd MLE example, the objective function is:

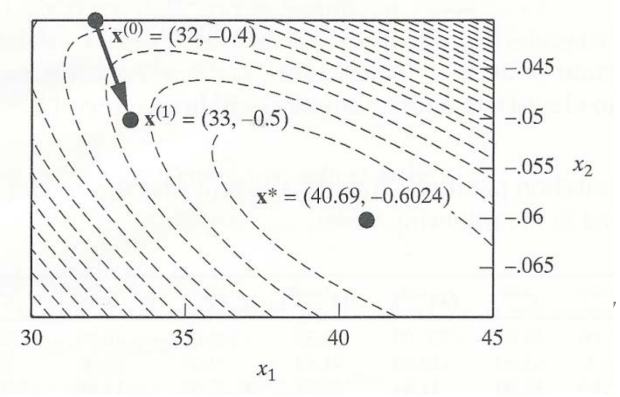
$$\max\ f(x_1, x_2) \triangleq \prod_{i=1}^{m} \left[\frac{\Gamma\left(x_1 + x_2\right)}{\Gamma\left(x_1\right)\Gamma\left(x_2\right)} \left(p_i\right)^{x_1 - 1} \left(1 - p_i\right)^{x_2 - 1} \right]$$

- Although derivatives exist in theory, they are not readily available, because the Γ-function does not have a closed form.
- To find the optimal value for the parameters, the search cannot be based on using closed form derivatives.

Improving Search Paradigm Revisited

- Recall that an improving search:
  - Starts at an initial solution $\mathbf{x}^{(0)}$
  - At each iteration t, advances to new solution $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \lambda\Delta\mathbf{x}$, where, where $\Delta\mathbf{x}$ is an improving and feasible direction, and λ is a step size
  - Repeats until no feasible directions can produce immediate improvements and a local optimum is reached

- The figure below shows for the nonlinear regression example, a search that starts at $\mathbf{x}^{(0)}$ and proceeds to an improved feasible solution $\mathbf{x}^{(1)}$.



First Derivatives & Gradients

- For single-variable functions, the first derivative $f'(x)$ provides information about the slope, or rate of change in function f for a small change in the value of x.
- Similarly, for functions with n variables, the ***gradient vector*** $\nabla f(\mathbf{x})$ provide the rate of change of f for small changes in each of the n variables:

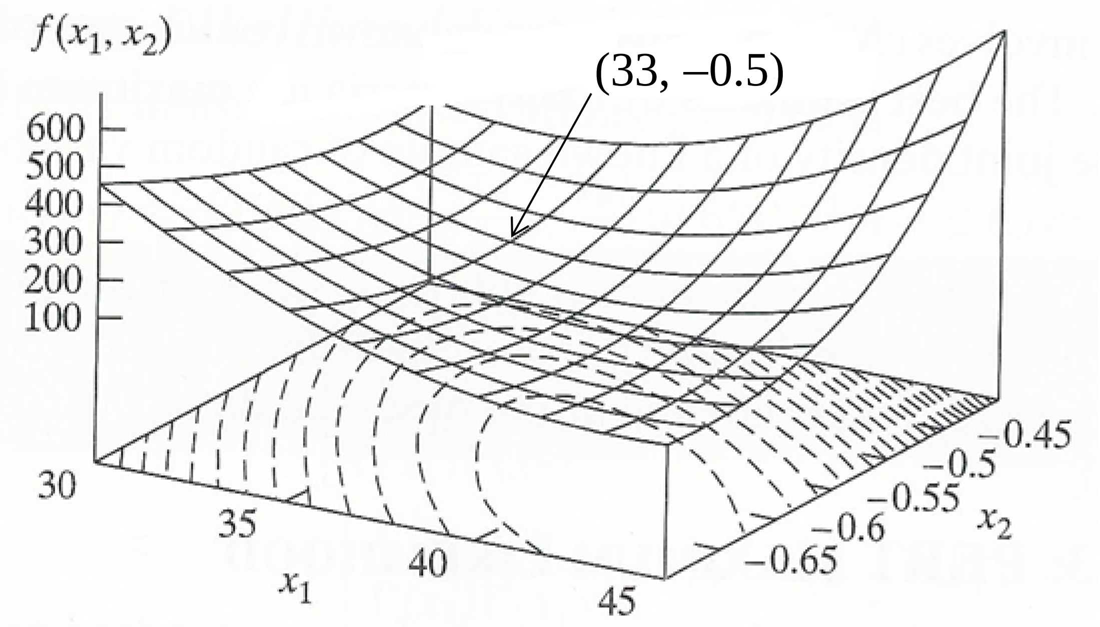
$$\nabla f^T(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_n} \end{pmatrix}$$

First Derivatives & Gradients

At point $\mathbf{x}^{(1)} = (33, -0.5)$ for the nonlinear regression example, it can be verified that:

$$\frac{\partial f}{\partial x_1} = -2\sum_{i=1}^{12}\left(q_i - 33p_i^{-0.5}\right)p_i^{-0.5} \approx -23.07$$

$$\frac{\partial f}{\partial x_2} = -2\sum_{i=1}^{12}\left(q_i - 33p_i^{-0.5}\right)\left(33p_i^{-0.5}\right)\ln\left(p_i\right) \approx -174.23$$



Rate of change is more rapid for small changes in x_2 than for x_1 as can be seen in figure.

Second Derivatives & Hessians

- For single-variable functions, the second derivative $f''(x)$ provides information about the rate of change of the slope, or the ***curvature***, of f.
- For functions with n variables, the ***Hessian*** matrix $\mathbf{H}(\mathbf{x})$ describes the rate of change of the gradient, or the curvature, of f in the neighborhood of the current solution:

$$\mathbf{H}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

For row i, column j the entry is $\frac{\partial^2 f}{\partial x_i \partial x_j}$.

Second Derivatives & Hessians – Cont'd

- For the nonlinear regression example, the expressions for the first derivatives were:

$$\frac{\partial f}{\partial x_1} = -2\sum_{i=1}^{12}\left(q_i - x_1 p_i^{x_2}\right) p_i^{x_2}, \text{ and}$$

$$\frac{\partial f}{\partial x_2} = -2\sum_{i=1}^{12}\left(q_i - x_1 p_i^{x_2}\right)\left(x_1 p_i^{x_2}\right)\ln\left(p_i\right)$$

- The expressions for the second partial derivatives are:

$$\frac{\partial^2 f}{\partial {x_1}^2} = 2 p_i^{2x_2}$$

$$\frac{\partial f}{\partial x_1 \partial x_2} = \frac{\partial f}{\partial x_2 \partial x_1} = -2\sum_{i=1}^{12}\left[\left(q_i - x_1 p_i^{x_2}\right)\left(p_i^{x_2}\right)\ln\left(p_i\right) - \left(p_i^{x_2}\right)\left(x_1 p_i^{x_2}\right)\ln\left(p_i\right)\right]$$

$$\frac{\partial f}{\partial {x_2}^2} = -2\sum_{i=1}^{12}\ln^2\left(p_i\right)\left[\left(q_i - x_1 p_i^{x_2}\right)\left(x_1 p_i^{x_2}\right) - \left(x_1 p_i^{x_2}\right)^2\right]$$

Second Derivatives & Hessians – Cont'd

- At point $\mathbf{x}^{(1)} = (33, -0.5)$ for the nonlinear regression example, it can be verified that:

$$\mathbf{H}(33,-0.5) = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x_1{}^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} \end{pmatrix} \approx \begin{pmatrix} 5.77 & 179.65 \\ 179.65 & 11{,}003.12 \end{pmatrix}$$

- The large value for $\dfrac{\partial^2 f}{\partial x_2{}^2}$ (11,003.12) in comparison to that of $\dfrac{\partial^2 f}{\partial x_1{}^2}$ (5.77) can be seen in the plot, where the rate of change in the slope near $\mathbf{x}^{(1)}$ is much steeper in the direction of x_2 than in that of x_1.

Taylor Series Approximation

One Variable

- The ***Taylor series*** approximation provides a more complete understanding of the change in the objective function in the neighborhood of the current solution.
- For one-variable functions, the impact on the function of a small change λ from the current solution $x^{(t)}$ is approximately:

$$f\left(x^{(t)}+\lambda\right) \approx f\left(x^{(t)}\right)+\frac{\lambda}{1!} f'\left(x^{(t)}\right)+\frac{\lambda^{2}}{2!} f''\left(x^{(t)}\right)+\frac{\lambda^{3}}{3!} f'''\left(x^{(t)}\right)+\cdots$$

where f' is the first derivative, f'' is the second derivative, and so on.

Taylor Series Approximation

One Variable – Cont'd

- For example, consider the function $f(x) \triangleq e^{3x-6}$.
- We have: $f'(x) = 3e^{3x\text{-}6}$, $f''(x) = 9e^{3x\text{-}6}$, $f'''(x) = 27e^{3x\text{-}6}$, and so on.
- The Taylor series approximation near $x^{(t)} = 2$ is:

$$f(2+\lambda) \approx f(2) + \frac{\lambda}{1!} f'(2) + \frac{\lambda^2}{2!} f''(2) + \frac{\lambda^3}{3!} f'''(2) + \cdots$$

$$= 1 + 3\lambda + \frac{9}{2}\lambda^2 + \frac{27}{6}\lambda^3 + \cdots$$

- Note that as $|\lambda| \to 0$, the terms involving the higher powers of λ approach zero the most rapidly.
- That is why in the immediate neighborhood of a current solution, the first few terms are sufficient to approximate the function.

Taylor Series Approximation

One Variable – Cont'd

- The ***first-order*** (or ***linear***) Taylor series approximation is:

$$f_1\left(x^{(t)}+\lambda\right) \approx f\left(x^{(t)}\right)+\lambda f'\left(x^{(t)}\right)$$

- For $f(x) \triangleq e^{3x-6}$ at $x^{(t)}=2,$ the first-order approximation is:

$$f_1(2+\lambda)=1+3\lambda$$

- The ***second-order*** (or ***quadratic***) Taylor series approximation is:

$$f_2(2+\lambda) \approx f(2)+\frac{\lambda}{1!}f'(2)+\frac{\lambda^2}{2!}f''(2)$$

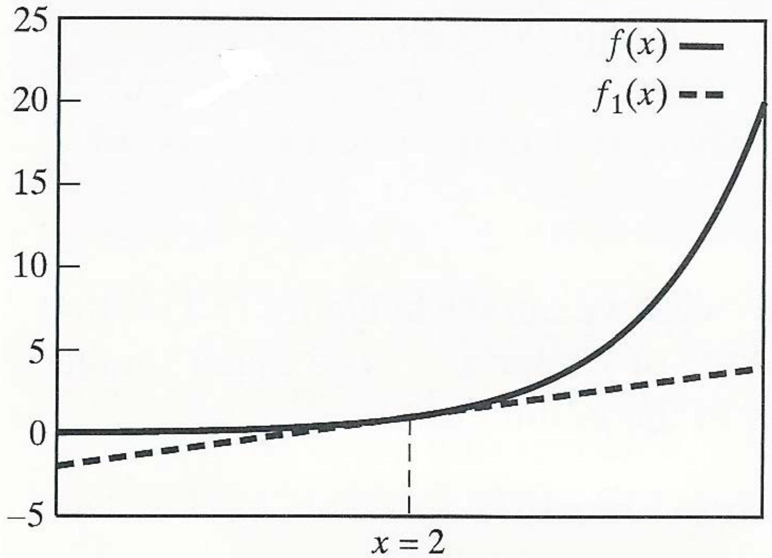
- For $f(x) \triangleq e^{3x-6}$ at $x^{(t)}=2,$ the second-order approximation is:

$$f_2(2+\lambda)=1+3\lambda+\frac{9}{2}\lambda^2$$

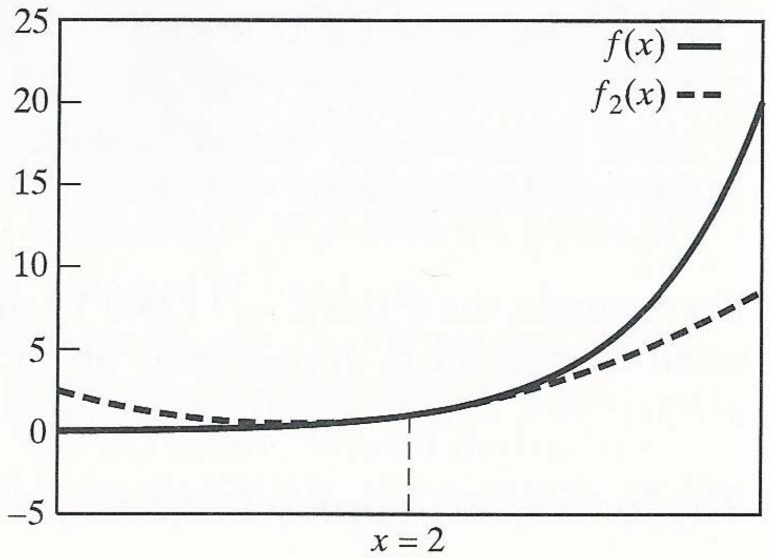
Taylor Series Approximation

One Variable – Cont'd

Graphically,



First-Order Approximation



Second-Order Approximation

Taylor Series Approximation

Multiple Variables

First-Order

- The first-order, or linear, approximation for the *n*-variable function $f(\mathbf{x}) = f(x_1, \ldots, x_n)$ at point $x^{(t)}$ is:

$$f_1\left(\mathbf{x}^{(t)} + \lambda\Delta\mathbf{x}\right) \triangleq f\left(\mathbf{x}^{(t)}\right) + \lambda\nabla f\left(\mathbf{x}^{(t)}\right)^T \cdot \Delta\mathbf{x}$$

$$= f\left(\mathbf{x}^{(t)}\right) + \lambda\sum_{j=1}^{n}\left(\frac{\partial f}{\partial x_j}\right)\Delta x_j$$

- For example, consider the function $f\left(x_1, x_2\right) \triangleq x_1 \ln\left(x_2\right) + 2$.
- What is the first-order approximation at $\mathbf{x}^{(t)} = (-3, 1)$?
- We have $f(-3, 1) = -3\times 0+2=2$.

Taylor Series Approximation

Multiple Variables

First-Order – Cont'd

- The gradient at $\mathbf{x}^{(t)} = (-3, 1)$ is:

$$\nabla f(-3,1) \triangleq \begin{pmatrix} \dfrac{\partial f}{x_1} \\ \dfrac{\partial f}{x_2} \end{pmatrix} = \begin{pmatrix} \ln(x_2) \\ \dfrac{x_1}{x_2} \end{pmatrix} = \begin{pmatrix} 0 \\ -3 \end{pmatrix}$$

- The first-order approximation is then:

$$\begin{aligned} f_1\left(\mathbf{x}^{(t)} + \lambda \Delta \mathbf{x}\right) &\triangleq f\left(\mathbf{x}^{(t)}\right) + \lambda \nabla f\left(\mathbf{x}^{(t)}\right)^T \cdot \Delta \mathbf{x} \\ &= 2 + \lambda(0,-3)\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} \\ &= 3 - 3\lambda \Delta x_2 \end{aligned}$$

Taylor Series Approximation

Multiple Variables

Second-Order

- The second-order, or quadratic, approximation for the n-variable function $f(\mathbf{x}) = f(x_1, \ldots, x_n)$ at point $\mathbf{x}^{(t)}$ is:

$$f_2\left(\mathbf{x}^{(t)} + \lambda\Delta\mathbf{x}\right) \triangleq f\left(\mathbf{x}^{(t)}\right) + \lambda\nabla f\left(\mathbf{x}^{(t)}\right)^T \cdot \Delta\mathbf{x} + \frac{\lambda^2}{2}\Delta\mathbf{x}^T \mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta\mathbf{x}$$

$$= f\left(\mathbf{x}^{(t)}\right) + \lambda\sum_{j=1}^{n}\left(\frac{\partial f}{\partial x_j}\right)\Delta x_j + \frac{\lambda^2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)\Delta x_i \Delta x_j$$

- For the same function $f\left(x_1, x_2\right) \triangleq x_1 \ln\left(x_2\right) + 2$, what is now the second-order approximation at $\mathbf{x}^{(t)} = (-3, 1)$?

Taylor Series Approximation

Multiple Variables

Second-Order – Cont'd

- First we compute the Hessian:

$$\mathbf{H}(-3,1)=\begin{pmatrix}\dfrac{\partial^2 f}{\partial {x_1}^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2}\\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2}\end{pmatrix}=\begin{pmatrix}0 & \dfrac{1}{x_2}\\ \dfrac{1}{x_2} & \dfrac{-x_1}{(x_2)^2}\end{pmatrix}=\begin{pmatrix}0 & 1\\ 1 & 3\end{pmatrix}$$

- The second order approximation is then:

$$f_2\left(\mathbf{x}^{(t)}+\lambda\Delta\mathbf{x}\right)\triangleq f\left(\mathbf{x}^{(t)}\right)+\lambda\nabla f\left(\mathbf{x}^{(t)}\right)^T\cdot\Delta\mathbf{x}+\frac{\lambda^2}{2}\Delta\mathbf{x}^T\mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta\mathbf{x}$$

$$=2+\lambda(0,-3)\begin{pmatrix}\Delta x_1\\ \Delta x_2\end{pmatrix}+\frac{\lambda^2}{2}(\Delta x_1,\Delta x_2)\begin{pmatrix}0 & 1\\ 1 & 3\end{pmatrix}\begin{pmatrix}\Delta x_1\\ \Delta x_2\end{pmatrix}$$

$$=2-3\lambda\Delta x_2+\lambda^2\Delta x_1\Delta x_2+\frac{3\lambda^2}{2}(\Delta x_2)^2$$

Stationary Points & Local Optima

- Solution $\mathbf{x}$ is a ***stationary point*** of a smooth function f if $\nabla f(\mathbf{x}) = 0$; i.e., all the partial derivatives equal 0.
- Consider the function:

$$f(x_1, x_2) \triangleq 40 + (x_1)^3 (x_1 - 4) + 3(x_2 - 5)^2$$

- The partial derivatives are:

$$\frac{\partial f}{\partial x_1} = (x_1)^3 + 3(x_1)^2 (x_1 - 4) = (x_1)^2 (4x_1 - 12)$$

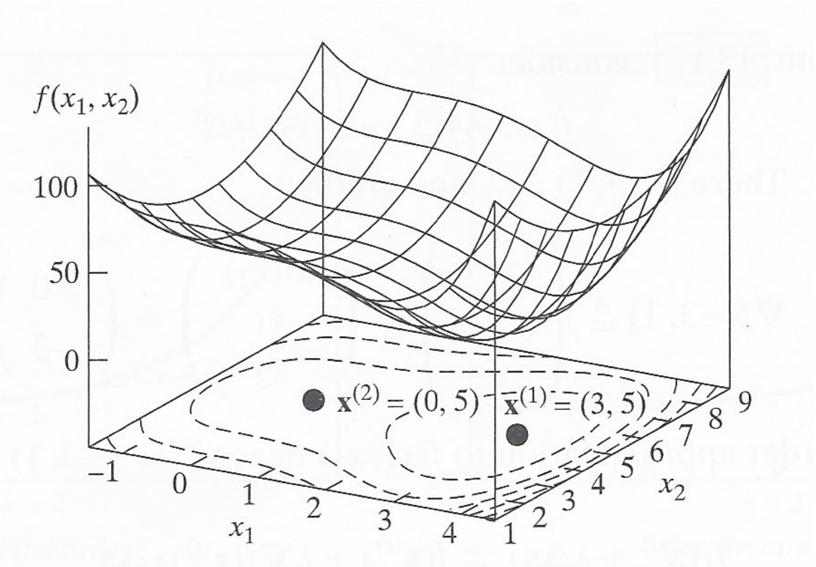
$$\frac{\partial f}{\partial x_2} = 6(x_2 - 5)$$

- There are two stationary points:

$$\mathbf{x}^{(1)} = (3,5), \text{ and } \mathbf{x}^{(2)} = (0,5).$$

Stationary Points & Local Optima – Cont'd

- From the plot below, it can be visually verified that $\mathbf{x}^{(1)}$ represents a local minimum:



- In general, ***any local optimum of an unconstrained smooth objective function is a stationary point***.

- This is referred to as ***first-order necessary*** condition for a local optimum.

Stationary Points & Local Optima – Cont'd

- To see why the statement is true, recall that if the gradient is chosen as the move direction, the instantaneous change in the objective function per unit step is approximately:

$$\nabla f\left(\mathbf{x}\right)^T . \nabla f\left(\mathbf{x}\right) = \sum_{j=1}^{n} \left(\frac{\partial f}{\partial x_j} \right)^2 \geq 0$$

- Hence, the gradient $\nabla f(\mathbf{x}^{(t)})$ is an improving direction for a maximize objective function, and its negative, $-\nabla f(\mathbf{x}^{(t)})$, is an improving direction for a minimize objective function.

Stationary Points & Local Optima – Cont'd

- The first-order Taylor approximations gives (+ for the maximization case, and – for the minimization case):

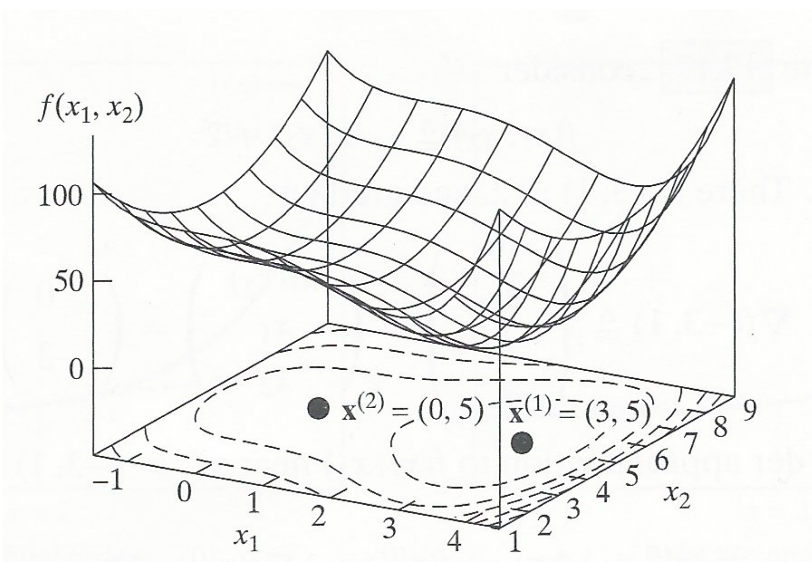
$$f_1\left(\mathbf{x}^{(t)} + \lambda\Delta\mathbf{x}\right) \approx f\left(\mathbf{x}^{(t)}\right) \pm \lambda\nabla f\left(\mathbf{x}^{(t)}\right)^T \cdot \nabla f\left(\mathbf{x}^{(t)}\right)$$

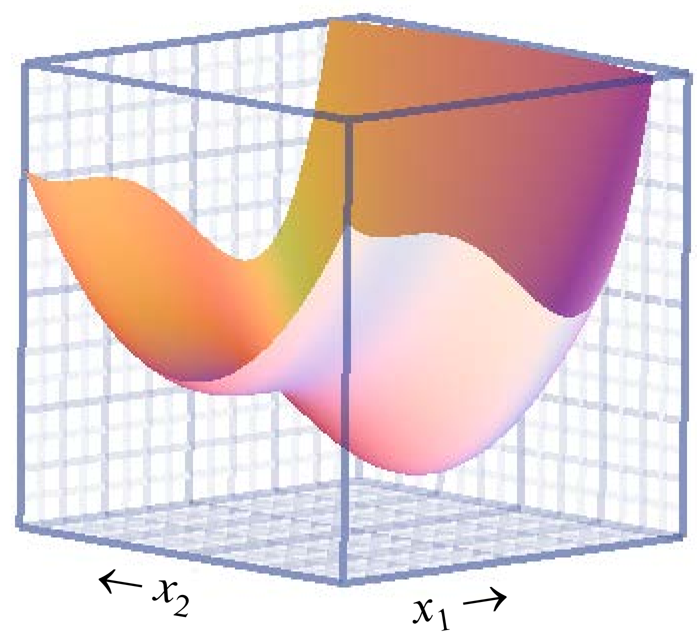
$$= f\left(\mathbf{x}^{(t)}\right) \pm \lambda\sum_{j=1}^{n}\left(\frac{\partial f}{\partial x_j}\right)^2$$

- It is known that for a "sufficiently small λ," the first order approximation dominates the subsequent terms.
- So, this is an improving direction, unless all the partial derivatives are zero, as required at a stationary point.

Saddle Points

- Stationary point $\mathbf{x}^{(2)} = (0, 5)$ in our example is a ***saddle point***:
  - Increasing x_1 reduces the objective.
  - Changing x_2 increases the objective.
- A saddle point is a stationary point that is neither a minimum nor a maximum.

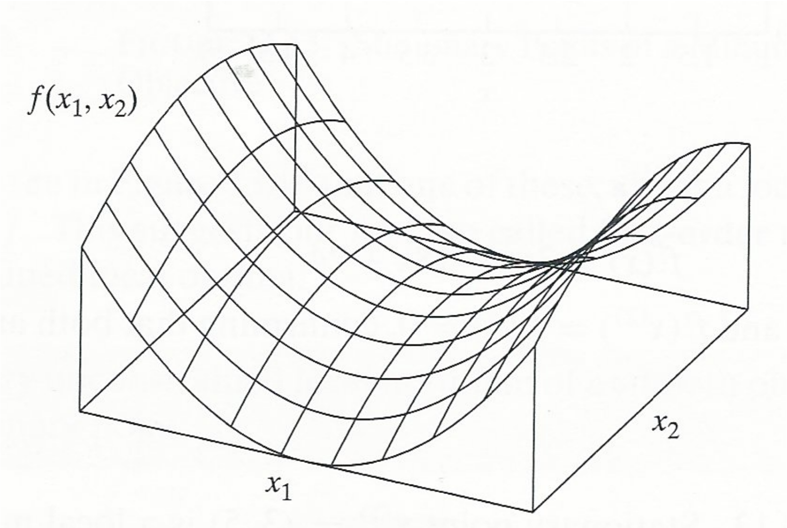




Another View of the Plot

Saddle Points – Cont'd

- Figure below shows a case that clearly looks like a saddle; hence, the name.
- In one dimension, the point is a local minimum, and in the other a local maximum.
- However, when *both* directions are considered, the point is neither a minimum nor a maximum.



Hessian Matrices and Local Optima

Preliminaries

- To distinguish between minima, maxima, and saddle points, it is necessary to look at the Hessian.
- At a stationary point, the second-order Taylor approximation is:

$$f_2\left(\mathbf{x}^{(t)} + \lambda\Delta\mathbf{x}\right) \approx f\left(\mathbf{x}^{(t)}\right) + \overbrace{\lambda\nabla f\left(\mathbf{x}^{(t)}\right)^T \cdot \Delta\mathbf{x}}^{0} + \frac{\lambda^2}{2}\Delta\mathbf{x}^T\mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta\mathbf{x}$$

$$= f\left(\mathbf{x}^{(t)}\right) + \frac{\lambda^2}{2}\Delta\mathbf{x}^T\mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta\mathbf{x}$$

- The *quadratic form* $\Delta\mathbf{x}^T\mathbf{H}(\mathbf{x}^{(t)})\Delta\mathbf{x}$ is critical in determining whether improving directions exist at $\mathbf{x}^{(t)}$.
- For example, if, for a minimization problem, we have $\Delta\mathbf{x}^T\mathbf{H}(\mathbf{x}^{(t)})\Delta\mathbf{x} < 0$ at a stationary point $\mathbf{x}^{(t)}$,for *some* direction $\Delta\mathbf{x}$, then $\mathbf{x}^{(t)}$ *cannot* be a local minimum.

Hessian Matrices and Local Optima

Necessary Conditions for Local Optima

- For any unconstrained local *maximum* of a smooth function f, the quadratic form $\Delta\mathbf{x}^T\mathbf{H}(\mathbf{x}^{(t)})\Delta\mathbf{x}$ is ≤ 0 for *any* $\Delta\mathbf{x} \neq 0$; i.e., the Hessian matrix $\mathbf{H}(\mathbf{x}^{(t)})$ has to be ***negative semidefinite***.

- Similarly, for any unconstrained local *minimum* of a smooth function f, the quadratic form $\Delta\mathbf{x}^T\mathbf{H}(\mathbf{x}^{(t)})\Delta\mathbf{x}$ is ≥ 0 for *any* $\Delta\mathbf{x} \neq 0$; i.e., the Hessian matrix $\mathbf{H}(\mathbf{x}^{(t)})$ has to be ***positive semidefinite***.

Hessian Matrices and Local Optima

Sufficient Conditions for Local Optima

- A stationary point of a smooth function f, is a local *maximum* if the quadratic form $\Delta\mathbf{x}^T\mathbf{H}(\mathbf{x}^{(t)})\Delta\mathbf{x}$ is < 0 for *any* $\Delta\mathbf{x} \neq 0$; i.e., if the Hessian matrix $\mathbf{H}(\mathbf{x}^{(t)})$ is ***negative definite***.

- Similarly, a stationary point of a smooth function f, is a local *minimum* if the quadratic form $\Delta\mathbf{x}^T\mathbf{H}(\mathbf{x}^{(t)})\Delta\mathbf{x}$ is > 0 for *any* $\Delta\mathbf{x} \neq 0$; i.e., if the Hessian matrix $\mathbf{H}(\mathbf{x}^{(t)})$ is ***positive definite***.

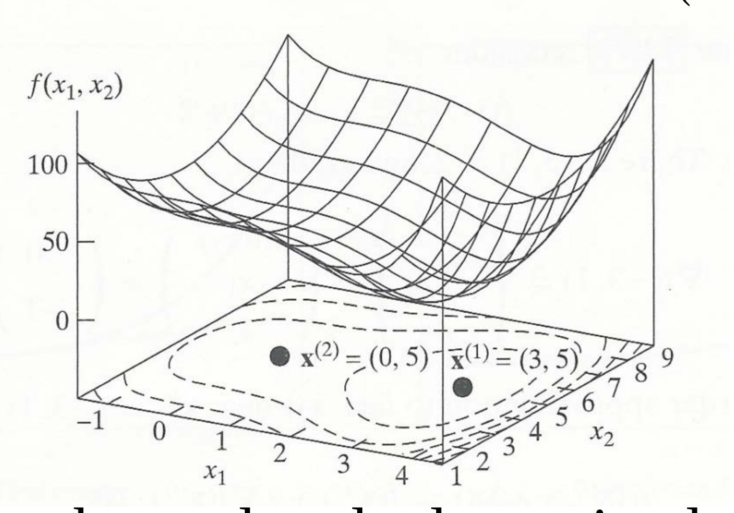
Hessian Matrices and Local Optima

Illustration of the Necessary Conditions

- Let's return to the example:

$$f(x_1, x_2) \triangleq 40 + (x_1)^3 (x_1 - 4) + 3(x_2 - 5)^2$$

with its two stationary points: $\mathbf{x}^{(1)} = (3,5)$, and $\mathbf{x}^{(2)} = (0,5)$.



- Visually, we have already determined that the stationary point $\mathbf{x}^{(1)}$ is a local minimum; let's verify that it satisfies the necessary conditions for it to be a local minimum.

Hessian Matrices and Local Optima

Illustration of the Necessary Conditions – Cont'd

- The partial derivatives are:

$$\frac{\partial f}{\partial x_1} = (x_1)^3 + 3(x_1)^2(x_1 - 4) = (x_1)^2(4x_1 - 12)$$

$$\frac{\partial f}{\partial x_2} = 6(x_2 - 5)$$

At $\mathbf{x}^{(1)} = (3,5)$, we have $\frac{\partial f}{\partial x_1} = 0$, and $\frac{\partial f}{\partial x_2} = 0$, and $\mathbf{x}^{(1)}$ is a stationary point.

- The Hessian at $\mathbf{x}^{(1)}$ = (3, 5) is:

$$\mathbf{H}(3,5) = \begin{pmatrix} 12(x_1)^2 - 24x_1 & 0 \\ 0 & 6 \end{pmatrix} = \begin{pmatrix} 36 & 0 \\ 0 & 6 \end{pmatrix}$$

Hessian Matrices and Local Optima

Illustration of the Necessary Conditions – Cont'd

- We can now evaluate $\Delta\mathbf{x}^T\mathbf{H}(\mathbf{x}^{(t)})\Delta\mathbf{x}$ for any direction $\Delta\mathbf{x} \neq \mathbf{0}$:

$$\Delta\mathbf{x}^T \begin{pmatrix} 36 & 0 \\ 0 & 6 \end{pmatrix} \Delta\mathbf{x} = \left(\Delta x_1, \Delta x_2\right) \begin{pmatrix} 36 & 0 \\ 0 & 6 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix}$$

$$= \left(36\Delta x_1, 6\Delta x_2\right) \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix}$$

$$= 36\left(\Delta x_1\right)^2 + 6\left(\Delta x_2\right)^2 > 0 \text{ for any } \Delta\mathbf{x} \neq 0$$

- This means that $\mathbf{H}(\mathbf{x}^{(t)})$ is positive definite, and, therefore, positive semidefinite, thus verifying the necessary conditions.

Hessian Matrices and Local Optima

Illustration of the Sufficient Conditions

- Let's now think of $\mathbf{x}^{(1)} = (3, 5)$ as a stationary point only, since we know that $\nabla f(\mathbf{x}^{(1)}) = 0$.
- Since the Hessian is positive definite, we can establish that $\mathbf{x}^{(1)}$ is a local minimum using the sufficient conditions.
- For the second stationary point, $\mathbf{x}^{(2)} = (0, 5)$, the Hessian is:

$$\mathbf{H}(0,5) = \begin{pmatrix} 12(x_1)^2 - 24x_1 & 0 \\ 0 & 6 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 6 \end{pmatrix}$$

- The quadratic form is:

$$\Delta\mathbf{x}^T \begin{pmatrix} 0 & 0 \\ 0 & 6 \end{pmatrix} \Delta\mathbf{x} = 6(\Delta x_2)^2 \geq 0, \text{for any } \Delta\mathbf{x} \neq 0$$

- Hence, the Hessian is *inconclusive* in this case as whether $\mathbf{x}^{(2)}$ is a saddle point or a local minimum. *Higher-order tests*, involving analyzing further terms in the Taylor's expansion, are necessary.

Hessian Matrices and Local Optima

Computational Method to Assess the Hessian

- One way to check for positive or negative definitiveness or semi-definitiveness of a symmetric matrix, is by calculating the determinants of its ***principal submatrices***; these are the submatrices made up of the first k rows and k columns, k = 1, …, n.
- A symmetric matrix is:
  - Positive definite if all the determinants are positive, and
  - Positive semidefinite if all the determinants are nonnegative.
- A symmetric matrix is:
  - Negative definite if all the determinants are nonzero, and alternating in sign, with the first one negative, and
  - Negative semidefinite if all the determinants are nonpositive and nonnegative, and alternating in sign, with the first one nonpositive.

Hessian Matrices and Local Optima

Computational Method to Assess the Hessian

Example 1

- Establish that **x** = (0, 0, 2) is a *local minimum* for the function:

$$f\left(x_1, x_2, x_3\right) \triangleq \left(x_1\right)^2 + x_1 x_2 + 5\left(x_2\right)^2 + 9\left(x_3 - 2\right)^2$$

- The partial derivatives are:

$$\frac{\partial f}{\partial x_1} = 2x_1 + x_2 \qquad \frac{\partial f}{\partial x_2} = x_1 + 10x_2 \qquad \frac{\partial f}{\partial x_3} = 18\left(x_3 - 2\right)$$

At **x** = (0, 0, 2), all the partial derivatives equal zero, and **x** is a stationary point.

- The Hessian at **x** = (0, 0, 2) is:

$$\mathbf{H}\left(\mathbf{x}\right) = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 10 & 0 \\ 0 & 0 & 18 \end{pmatrix}$$

Hessian Matrices and Local Optima

Computational Method to Assess the Hessian

Example 1 – Cont'd

- The determinants of the submatrices are:

$$\det(2) = 2 > 0$$

$$\det\begin{pmatrix} 2 & 1 \\ 1 & 10 \end{pmatrix} = 19 > 0$$

$$\det\begin{pmatrix} 2 & 1 & 0 \\ 1 & 10 & 0 \\ 0 & 0 & 18 \end{pmatrix} = 342 > 0$$

- Therefore, the Hessian is positive definite, and **x** is a local minimum.

Hessian Matrices and Local Optima

Computational Method to Assess the Hessian

Example 2

- Establish that **x** = (1, 0) is a *saddle* point for the function:

$$f(x_1, x_2) \triangleq (x_1)^2 - 2x_1 - (x_2)^2$$

- The partial derivatives are:

$$\frac{\partial f}{\partial x_1} = 2x_1 - 2 \qquad \frac{\partial f}{\partial x_2} = -2x_2$$

At **x** = (1, 0), the partial derivatives equal zero, and **x** is a stationary point.

- The Hessian at **x** = (1, 0) is:

$$\mathbf{H}(\mathbf{x}) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$$

- Checking the determinants of the submatrices:

$$\det(2) = 2 \text{ and } \det\begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix} = -4$$

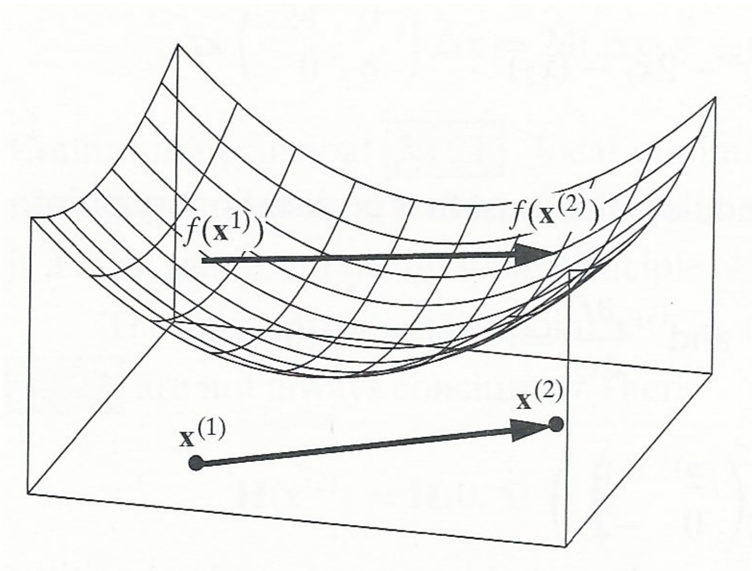
The Hessian is neither positive nor negative semidefinite, and cannot be a local minimum or maximum. **x** has then to be a saddle point.

Convexity/Concavity & Global Optimality

Convexity

A function f is ***convex*** if given any two points $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ in its domain, and any $\lambda \in [0, 1]$, we have:

$$f\left(\mathbf{x}^{(1)} + \lambda\left(\mathbf{x}^{(2)} - \mathbf{x}^{(1)}\right)\right) \leq f\left(\mathbf{x}^{(1)}\right) + \lambda\left(f\left(\mathbf{x}^{(2)}\right) - f\left(\mathbf{x}^{(1)}\right)\right)$$



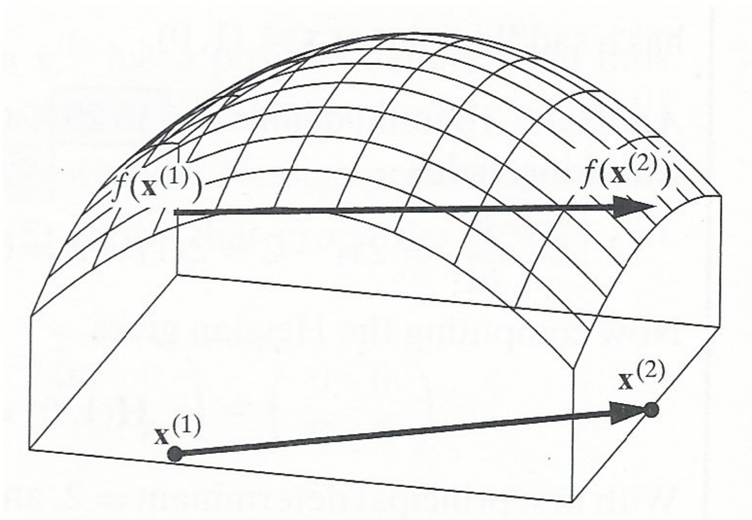
- $\mathbf{x}^{(1)} + \lambda(\mathbf{x}^{(2)} - \mathbf{x}^{(1)})$, with $\lambda \in (0, 1)$ is the trajectory of all points along the direction $(\mathbf{x}^{(2)} - \mathbf{x}^{(1)})$.
- The interpolation, $f(\mathbf{x}^{(1)}) + \lambda(f(\mathbf{x}^{(2)}) - f(\mathbf{x}^{(1)}))$ of the function value along the trajectory should always exceed or equal the true function value $f(\mathbf{x}^{(1)} + \lambda(\mathbf{x}^{(2)} - \mathbf{x}^{(1)}))$.

Convexity/Concavity & Global Optimality

Concavity

A function f is ***concave*** if given any two points $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ in its domain, and any $\lambda \in [0, 1]$, we have:

$$f\left(\mathbf{x}^{(1)}+\lambda\left(\mathbf{x}^{(2)}-\mathbf{x}^{(1)}\right)\right) \geq f\left(\mathbf{x}^{(1)}\right)+\lambda\left(f\left(\mathbf{x}^{(2)}\right)-f\left(\mathbf{x}^{(1)}\right)\right)$$

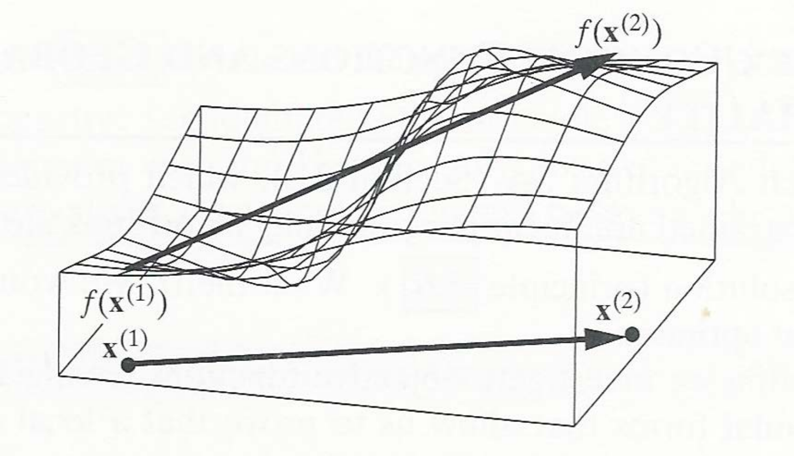


The interpolation, $f(\mathbf{x}^{(1)})+\lambda(f(\mathbf{x}^{(2)})- f(\mathbf{x}^{(1)}))$ of the function value along the trajectory should always fall below or equal the true function value $f(\mathbf{x}^{(1)} + \lambda(\mathbf{x}^{(2)} - \mathbf{x}^{(1)}))$.

Convexity/Concavity & Global Optimality

Convexity and Concavity

- Functions may be ***neither convex or concave***:



The function meets neither of the two definitions.

- Linear functions are ***both convex and concave*** as they satisfy both of the definitions.

Convexity/Concavity & Global Optimality

Sufficient Conditions for Global Optima

If $f(\mathbf{x})$ is a concave function, then any unconstrained local maximum is an unconstrained global maximum, and if $f(\mathbf{x})$ is a convex function, then any unconstrained local minimum is an unconstrained global minimum.

Why?

- Consider a convex function $f(\mathbf{x})$ with a global minimum $\mathbf{x}^*$, and *any* other solution $\mathbf{x}^{(1)}$ that is not a global minimum; i.e., we have $f(\mathbf{x}^*) < f(\mathbf{x}^{(1)})$, and $\lambda(f(\mathbf{x}^*) - f(\mathbf{x}^{(1)})) < 0$ for any $\lambda > 0$.
- Using convexity, and the fact that $\lambda(f(\mathbf{x}^*) - f(\mathbf{x}^{(1)})) < 0$ we have:

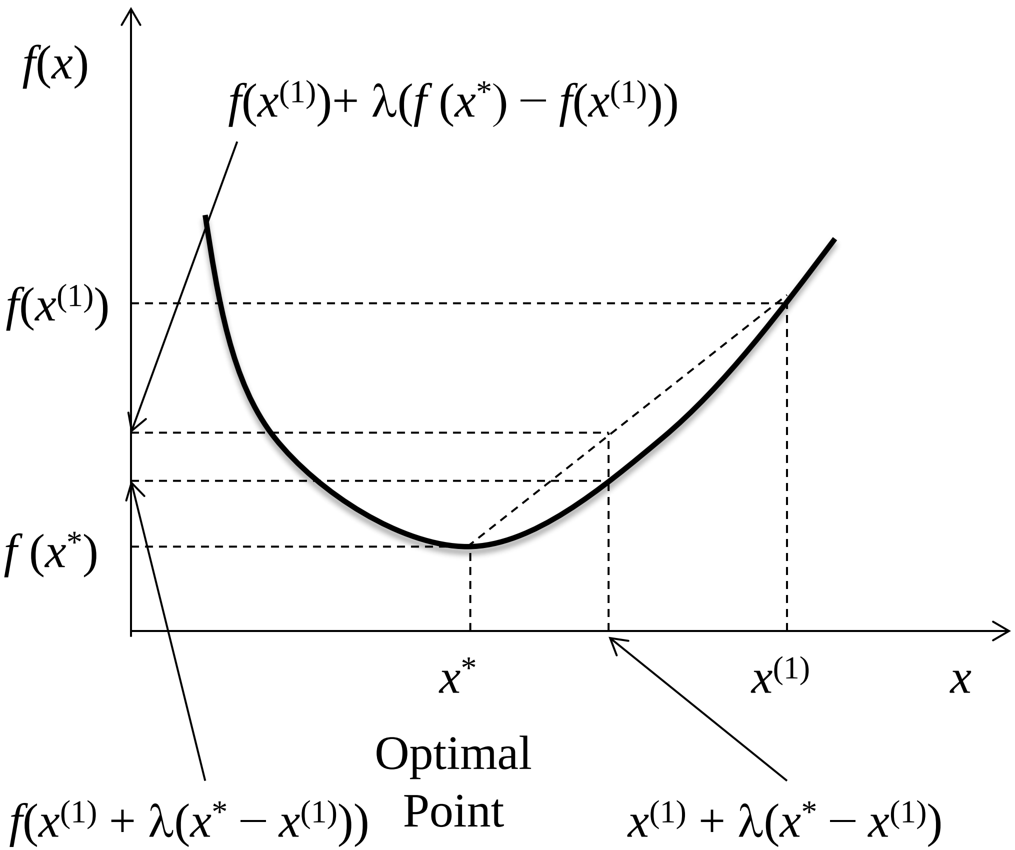
$$f\left(\mathbf{x}^{(1)} + \lambda\left(\mathbf{x}^* - \mathbf{x}^{(1)}\right)\right) \le f\left(\mathbf{x}^{(1)}\right) + \underbrace{\lambda\left(f\left(\mathbf{x}^*\right) - f\left(\mathbf{x}^{(1)}\right)\right)}_{<0} < f\left(\mathbf{x}^{(1)}\right), \text{ for } \textit{any } \lambda > 0$$

- This means that $\Delta\mathbf{x} = \mathbf{x}^* - \mathbf{x}^{(1)}$ is an improving direction at $\mathbf{x}^{(1)}$ and no local minimum can exist.

Convexity/Concavity & Global Optimality

Sufficient Conditions for Global Optima

Illustration Using a Single-Variable Convex Function

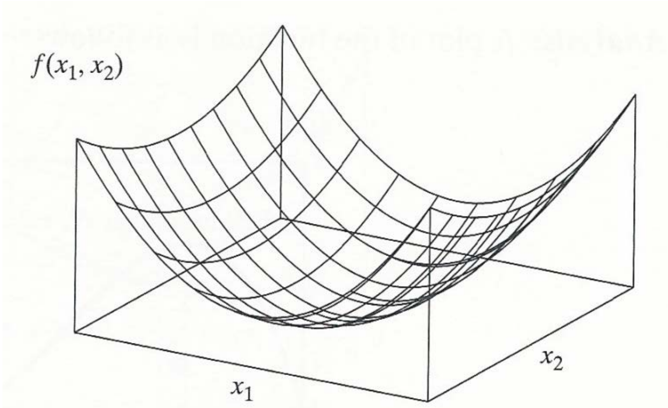


- x^* is the global optimum, and $x^{(1)}$ is any other point.
- Because the function is convex, the objective function value $f(x^{(1)} + \lambda(x^* - x^{(1)}))$, for the point $x^{(1)} + \lambda(x^* - x^{(1)})$, lies *below* $f(x^{(1)}) + \lambda(f(x^*) - f(x^{(1)}))$; i.e., $f(x^{(1)} + \lambda(x^* - x^{(1)})) \leq f(x^{(1)}) + \lambda(f(x^*) - f(x^{(1)}))$.
- Also, because x^* is a global optimum, and $x^{(1)}$ is not, we have $\lambda(f(x^*) - f(x^{(1)})) < 0$ for any $\lambda > 0$.
- Therefore, $f(x^{(1)} + \lambda(x^* - x^{(1)}))$ *must be* $< f(x^{(1)})$ for any $\lambda > 0$, and $x^{(1)}$ cannot be a local minimum.

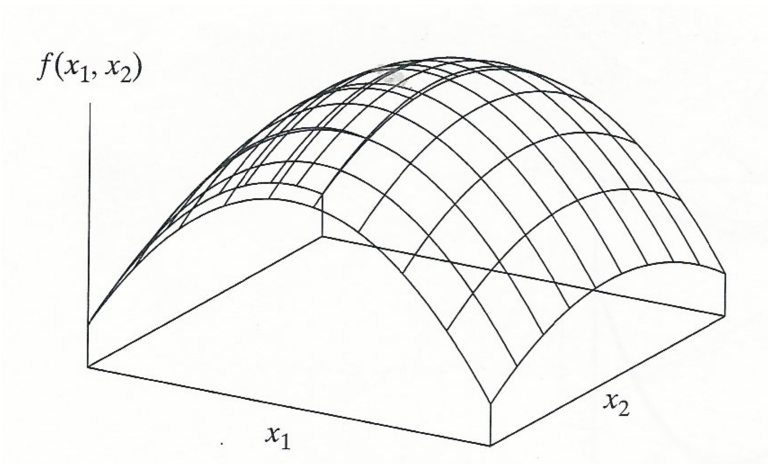
Convexity/Concavity & Global Optimality

Sufficient Conditions for Global Optima

The Sufficient Conditions are "Obvious" in the 3-D Plots



A unique global minimum exists



A unique global maximum exists

Convexity/Concavity & Global Optimality

Stationary Points & Convexity/Concavity

Every stationary point of a *smooth* concave function is an unconstrained global maximum, and every stationary point of a *smooth* convex function is an unconstrained global minimum.

Why?

- One of the important convexity/concavity results is that:
  - f is convex if and only if $\mathbf{H}(\mathbf{x})$ is positive semidefinite for all $\mathbf{x}$ in its domain, and
  - f is concave if and only if $\mathbf{H}(\mathbf{x})$ is negative semidefinite for all $\mathbf{x}$ in its domain.

- For single-variable functions:
  - f is convex if and only $f''(x) \geq 0$ for all x in its domain, and
  - f is concave if and only $f''(x) \leq 0$ for all x in its domain.

Convexity/Concavity & Global Optimality

Stationary Points & Convexity/Concavity

Why? – Cont'd

- For a stationary point x^* of a smooth convex function, Taylor's second-order approximation gives:

$$f_2\left(\mathbf{x}^{(t)}+\lambda\Delta\mathbf{x}\right) \triangleq f\left(\mathbf{x}^{(t)}\right)+\lambda\underbrace{\nabla f\left(\mathbf{x}^{(t)}\right)^T}_{0}\cdot\Delta\mathbf{x}+\frac{\lambda^2}{2}\Delta\mathbf{x}^T\mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta\mathbf{x}$$

$$= f\left(\mathbf{x}^{(t)}\right)+\frac{\lambda^2}{2}\underbrace{\Delta\mathbf{x}^T\mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta\mathbf{x}}_{\geq 0} \geq f\left(\mathbf{x}^{(t)}\right)$$

- So point x^* is a local minimum.
- However, we've already seen that any local minimum for a convex function is also the global minimum, and, so, the stationary point x^* of the convex function is a global minimum.
- A similar argument can show that any stationary point of a smooth concave function is a global maximum.

Tests for Convexity/Concavity

1. If $f(\mathbf{x})$ is convex, then $-f(\mathbf{x})$ is concave, and vice versa.
2. A function f is convex if and only if $\mathbf{H}(\mathbf{x})$ is positive semidefinite for all $\mathbf{x}$ in its domain, and it is concave if and only if $\mathbf{H}(\mathbf{x})$ is negative semidefinite for all $\mathbf{x}$ in its domain.
3. Linear functions are both convex and concave.
4. A nonnegative combination of convex functions is convex, and a nonnegative combination of concave functions is concave:

$$f(\mathbf{x}) \triangleq \sum_{i=1}^{k} \underbrace{\alpha_i}_{\alpha_i \geq 0} g_i(\mathbf{x})$$

- If $g_i(\mathbf{x})$, $i = 1, .., k$ is convex, then $f(\mathbf{x})$ is convex.
- If $g_i(\mathbf{x})$, $i = 1, .., k$ is concave, then $f(\mathbf{x})$ is concave.

Tests for Convexity/Concavity – Cont'd

5. The *maximum* of *convex* functions is convex, and the *minimum* of *concave* functions is concave:

$$f(\mathbf{x}) \triangleq \max\left(g_i(\mathbf{x}) : i = 1,\ldots,k\right)$$ If $g_i(\mathbf{x})$, i =1, .., k is convex, the $f(\mathbf{x})$ is convex

$$f(\mathbf{x}) \triangleq \min\left(g_i(\mathbf{x}) : i = 1,\ldots,k\right)$$ If $g_i(\mathbf{x})$, i =1, .., k is concave, the $f(\mathbf{x})$ is concave

6. Let $h(\mathbf{x})$ denote a multiple-variable function in $\mathbf{x}$, and $g(y)$ a *non-decreasing* single-variable function in y:
   – If $h(\mathbf{x})$ and $g(y)$ are convex, then $g(h(\mathbf{x}))$ is convex.
   – If $h(\mathbf{x})$ and $g(y)$ are concave, then $g(h(\mathbf{x}))$ is concave

7. If $g(\mathbf{x})$ is concave, then $f(\mathbf{x}) \triangleq 1/g(\mathbf{x})$ is convex over $\mathbf{x}$ such that $g(\mathbf{x}) > 0$, and if $g(\mathbf{x})$ is convex, then $f(\mathbf{x}) \triangleq 1/g(\mathbf{x})$ is concave over $\mathbf{x}$ such that $g(\mathbf{x}) < 0$.

Tests for Convexity/Concavity

Example 1

- Consider the curve-fitting objective function for linear regression:

$$\min f\left(x_1, x_2\right) \triangleq \sum_{i=1}^{m}\left[q_i - \left(x_1 + x_2 p_i\right)\right]^2$$

- Now, $q_i-(x_1+x_2p_i)$ is linear, and, hence, convex (Rule 3).
- The single-variable function $g(y) = y^2$ is has $g''(y) = 2$, and is then convex (Rule 2).
- The single-variable function $g(y) = y^2$ is also nondecreasing over the domain $y \geq 0$.
- Hence $[q_i-(x_1+x_2p_i)]^2$ is then convex (Rule 6).
- The objective function, being the sum of convex terms, is then itself convex (Rule 4).

Tests for Convexity/Concavity

Example 2

- Consider the function:

$$f(x_1, x_2) \triangleq (x_1+1)^4 + x_1 x_2 + (x_2+1)^4 \text{ over all } x_1, x_2 \geq 0$$

- The gradient and Hessian of the function are :

$$\nabla f(x_1, x_2) = \begin{pmatrix} 4(x_1+1)^3 + x_2 \\ x_1 + 4(x_2+1)^3 \end{pmatrix}, \text{ and } \mathbf{H}(x_1, x_2) = \begin{pmatrix} 12(x_1+1)^2 & 1 \\ 1 & 12(x_2+1)^2 \end{pmatrix}$$

- The principal determinants of the Hessian function are:

$$12(x_1+1)^2 \text{ and } 144(x_1+1)^2(x_2+1)^2 - 1$$

- Both principal determinants are > 0 for all x_1 and $x_2 \geq 0$; therefore, the Hessian is positive definite, and the function is convex (Rule 2).

Tests for Convexity/Concavity

Example 3

- Consider the function:

  $f\left(x_1, x_2\right) \triangleq e^{-3x^1+x^2}$ over all x_1, x_2

- The function $-3x_1 + x_2$ is linear, and, hence, both convex and concave (Rule 3).

- The single-variable function $g(y) = e^y$ is nondecreasing; it also has $g''(y) = e^y > 0$ for any y, and is then convex (Rule 2).

- $g(-3x_1 + x_2) = e^{(-3x1 + x2)}$ is then convex (Rule 6).

Tests for Convexity/Concavity

Example 4

- Consider the function:

$$f\left(x_1, x_2, x_3\right) \triangleq -4\left(x_1\right)^2 + 5x_1x_2 - 2\left(x_2\right)^2 + 18x_3 \text{ over all } x_1, x_2$$

- The gradient and Hessian of the function are :

$$\nabla f\left(x_1, x_2, x_3\right) = \begin{pmatrix} -8x_1 + 5x_2 \\ 5x_1 - 4x_2 \\ 18 \end{pmatrix}, \text{ and } \mathbf{H}\left(x_1, x_2, x_3\right) = \begin{pmatrix} -8 & 5 & 0 \\ 5 & -4 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- The determinants of the principal submatrices are –8, 7, and 0; therefore, the Hessian is negative semi-definite, and the function is concave (Rule 2).

Tests for Convexity/Concavity

Example 5

- Consider the function:

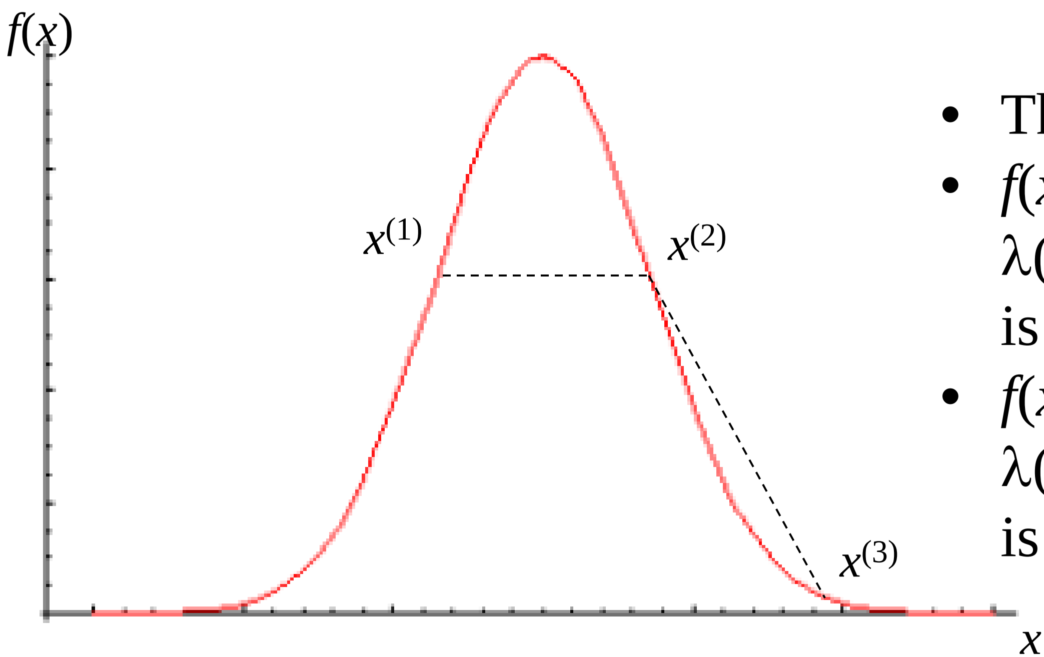
  $$f\left(x_1, x_2\right) \triangleq \frac{1}{-7x_1} - e^{-3x_1 + x_2}, \text{ over all } x_1, x_2 > 0$$

- The second term is the negative of a convex function, and is therefore concave (Rule 1).
- The denominator of the first term is linear (both convex and concave per Rule 3) and negative over the function's domain; therefore, its reciprocal is concave (Rule 7).
- Since the sum of concave functions is concave (Rule 4), the function is concave.

Unimodality, Convexity and Concavity

- Recall that an objective function $f(\mathbf{x})$ is ***unimodal*** if for every $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ such $f(\mathbf{x}^{(2)})$ is superior to $f(\mathbf{x}^{(1)})$ (> for a maximize and < for a minimize), direction $\Delta\mathbf{x}=(\mathbf{x}^{(2)} - \mathbf{x}^{(1)})$ is improving at $\mathbf{x}^{(1)}$.
- Convex and concave functions are unimodal because, as we've seen, for such functions, improving directions exist for all points other than the global optimum (see slide #56).
- However, a unimodal function is more general, and can be neither convex nor concave.
- *See next slide…*

Unimodality, Convexity and Concavity – Cont'd



- The function is unimodal.
- $f(x^{(1)} + \lambda(x^{(2)} - x^{(1)}) > f(x^{(1)}) + \lambda(f(x^{(2)} - f(x^{(1)})$, and the function is not convex.
- $f(x^{(2)} + \lambda(x^{(3)} - x^{(2)}) < f(x^{(2)}) + \lambda(f(x^{(3)} - f(x^{(2)})$, and the function is not concave either.

- No comparable set of rules exist for unimodal functions as those that exist for convex and concave ones.
- So, typically, we test for convexity/concavity and not for unimodality.

Algorithms for Unconstrained Optimization

One-Dimensional Search

- ***One-dimensional search*** algorithms are used for single-variable NLPs:
  - Such NLPs may occur in certain applications.
  - More commonly, the algorithms are used in line searches to determine step sizes within more general NLP algorithms.
- One-dimensional searches typically do not use derivatives:
  - ***Golden section search***
  - ***Quadratic fit search***
- For unimodal functions, ***bracketing methods*** are used to determine a search range prior to invoking the one-dimensional search algorithm.

Algorithms for Unconstrained Optimization

Differentiable Functions

Gradient Search - Overview

- ***Gradient search*** uses the first-order Taylor's approximation to select improving directions at point $\mathbf{x}^{(t)}$:
  - $\Delta\mathbf{x} \triangleq +\nabla f\left(\mathbf{x}^{(t)}\right)$, for a maximize function
  - $\Delta\mathbf{x} \triangleq -\nabla f\left(\mathbf{x}^{(t)}\right)$, for a minimize function

- At each iteration, a 1-dimensional search is used to determine the value of λ that optimizes:

$$\max(\text{or } \min)\, f(\mathbf{x}^{(t)} + \lambda\Delta\mathbf{x})$$

- The algorithm stops when the *magnitude* of the gradient vector, or the ***gradient norm***, falls below a pre-specified level, ε :

$$\left\|\nabla f\left(x^{(t)}\right)\right\| \triangleq \sqrt{\sum_{j=1}^{n}\left(\frac{\partial f}{\partial x_j}\right)^2} \leq \varepsilon$$

Algorithms for Unconstrained Optimization

Differentiable Functions

Gradient Search – Example

- For the nonlinear regression example, let's say that search initialized with the point: :

$$\mathbf{x}^{(0)} = (32.00, -0.4000)$$

- It can be verified that:

$$\frac{\partial f}{\partial x_1} = -2\sum_{i=1}^{12}\left(q_i - 32\,p_i^{-0.4}\right)p_i^{-0.4} \approx -6.24$$

$$\frac{\partial f}{\partial x_2} = -2\sum_{i=1}^{12}\left(q_i - 32\,p_i^{-0.4}\right)\left(32\,p_i^{-0.4}\right)\ln\left(p_i\right) \approx 1053.37$$

- The search direction will then be:

$$\Delta\mathbf{x} = -\nabla f^{T}(\mathbf{x}) = (6.24, -1053.37)$$

Algorithms for Unconstrained Optimization

Differentiable Functions

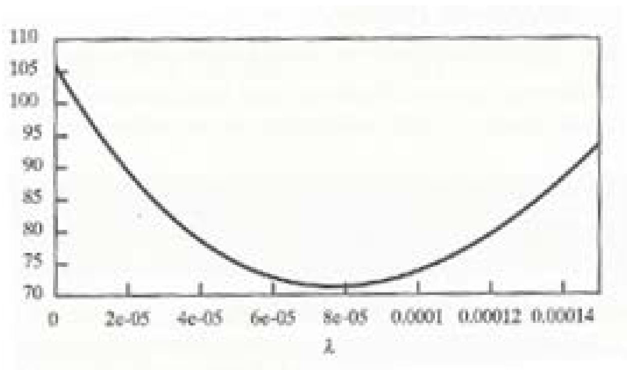
Gradient Search – Example – Cont'd

- The following 1-dimensional problem then needs to be solved:

$$\min f(\mathbf{x}^{(t)} + \lambda \Delta \mathbf{x}) = f(32 + 6.24\lambda, -0.4 - 1053.37\lambda)$$

$$= \sum_{i=1}^{m} \left[q_i - (32 + 6.24\lambda)\, p_i^{-0.4-1053.37\lambda} \right]^2$$

- The following is the plot for this convex 1-dimensional function:

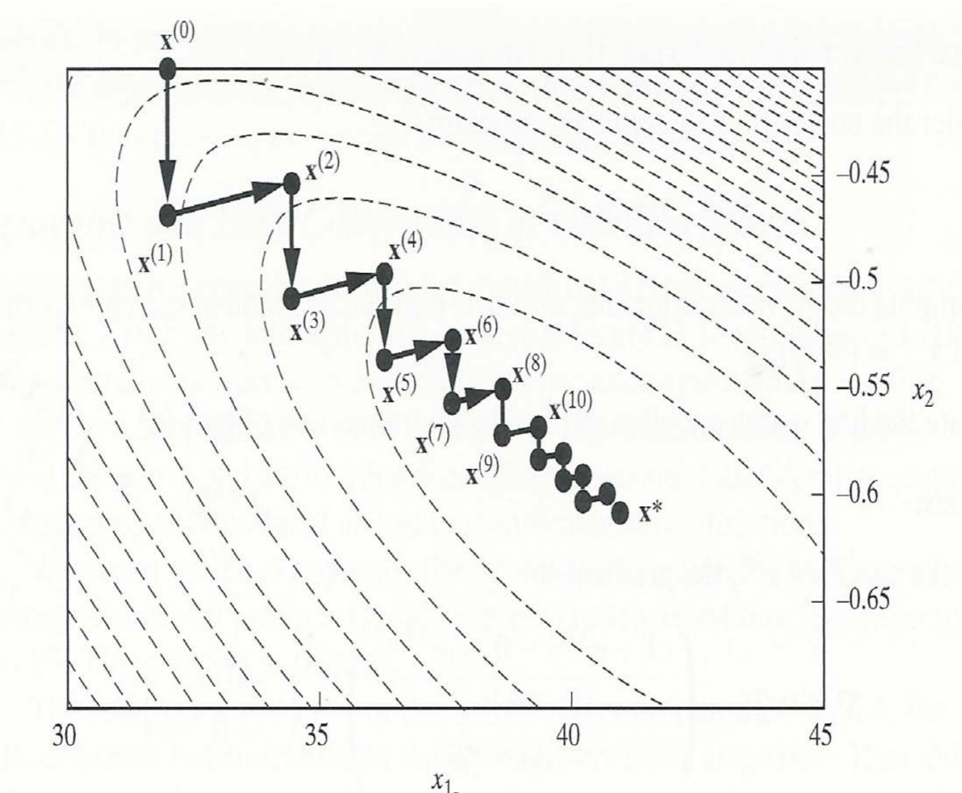


– The minimum occurs at approximately $\lambda_1 = 0.00007$.
– The new point is $x^{(1)} \leftarrow x^{(0)} + \lambda_1$ and the new $\mathbf{x} \approx (32, -0.4687)$.

Algorithms for Unconstrained Optimization

Differentiable Functions

Gradient Search – Pros & Cons



Gradient Search for the Nonlinear Regression Example

- Gradient search produces the steepest local rate of improvement that is always tangential to the contour lines:
  - Gradients provides steepest ascent for a maximize function, and
  - Negative of the gradients provides steepest descent for a minimize function
- On the downside, is that the search exhibits zigzagging and poor convergence as the local optimal solution is approached.

Algorithms for Unconstrained Optimization

Differentiable Functions

Newton Method

- ***Newton method*** uses the second-order Taylor's approximation to select improving directions at point $\mathbf{x}^{(t)}$:

$$f_2\left(\mathbf{x}^{(t)}+\lambda\Delta\mathbf{x}\right)\approx f\left(\mathbf{x}^{(t)}\right)+\lambda\nabla f\left(\mathbf{x}^{(t)}\right)^T\cdot\Delta\mathbf{x}+\frac{\lambda^2}{2}\Delta\mathbf{x}^T\mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta\mathbf{x}$$

$$=f\left(\mathbf{x}^{(t)}\right)+\lambda\sum_{j=1}^{n}\left(\frac{\partial f}{\partial x_j}\right)\Delta x_j+\frac{\lambda^2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial^2 f}{\partial x_i\partial x_j}\right)\Delta x_i\Delta x_j$$

- Instead of specifying a step size, the method sets $\lambda = 1$, and determines a move towards the local optimum of the 2nd order approximation.
- Setting λ=1 and taking partial derivatives with respect to the move components:

$$\frac{\partial f_2}{\partial\Delta x_i}=\left(\frac{\partial f}{\partial x_i}\right)+\sum_{j=1}^{n}\left(\frac{\partial^2 f}{\partial x_i\partial x_j}\right)\Delta x_j,\, i=1,\cdots,n$$

Algorithms for Unconstrained Optimization

Differentiable Functions

Newton Method – Cont'd

- In matrix form,

The Hessian of f at current point $x^{(t)}$.

$$\nabla f_2\left(\Delta \mathbf{x}\right) = \nabla f\left(\mathbf{x}^{(t)}\right) + \mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta \mathbf{x}$$

Gradient for the 2nd-order Taylor's approximation with respect to the move direction.

The gradient of f at current point $x^{(t)}$.

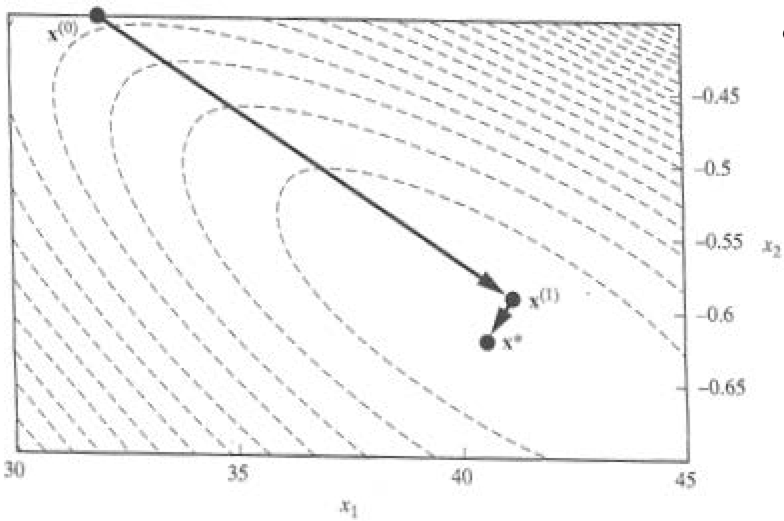
- Setting $\nabla f_2(\Delta \mathbf{x})=\mathbf{0}$, gives an $n\times n$ system of equations that can be used to solve for the ***Newton step*** ($\Delta \mathbf{x}$) used at each iteration:

$$\mathbf{H}\left(\mathbf{x}^{(t)}\right)\Delta \mathbf{x} = -\nabla f\left(\mathbf{x}^{(t)}\right)$$

Algorithms for Unconstrained Optimization

Differentiable Functions

Newton Method – Cont'd



- The search converges to the local optimum in dramatically fewer steps.
- Computational tradeoffs per iteration:
  - Both first and second derivatives are now needed.
  - In addition, a system of equations has to be solved for the move direction.
  - However, the 1-dimensional search to find the step λ for the move is not needed.
- Another difficulty is that Newton search is assured to converge to the local optimum only if the it starts "relatively close" it.

Algorithms for Unconstrained Optimization

Differentiable Functions

Quasi-Newton Methods

- ***Quasi-Newton methods*** provide the most effective algorithms for unconstrained nonlinear programs.
- These methods seek to avoid the:
  - Poor numerical performance of gradient methods, and
  - Expensive evaluation of Hessians and solution of system of equations required by the Newton method.
- Instead of using Hessians, quasi-Newton methods use "deflection matrices" that approximate the Hessians, and that can be updated efficiently.
- One-dimensional line searches are required to determine the step sizes.

Algorithms for Unconstrained Optimization

Nonsmooth Functions

- ***Nondifferentiable*** (also referred to as ***nonsmooth***) optimization refers to MPs where the objective and/or one or more of the constraints are nonsmooth.
- Gradients do not exist and the functions may have discontinuities and/or kinks or corner points.
- Specialized non-derivative based theory and methods have been developed (e.g. Nelder-Mead).