Multivariable Non Linear Programming

Multivariable Non linear Optimization

Consider a function f (x) where x is the n-vector $x = [x_1, x_2, . . . , x_n]^T$.

The gradient vector of this function is given by the partial derivatives with respect to each of the independent variables,

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \cdots \\ \cdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

In the multivariate case, the gradient vector is perpendicular to the the hyperplane tangent to the contour surfaces of f.

Multivariable Non linear Optimization

Example

$$f = 15x_1 + 2(x_2)^3 - 3x_1(x_3)^2$$

$$\Downarrow$$

$$\nabla f = \begin{bmatrix} 15 - 3(x_3)^2 & 6(x_2)^2 & -6x_1x_3 \end{bmatrix}$$

Multivariable Non linear Optimization

While the gradient of a function of n variables is an n-vector, the “second derivative” of an n-variable function is defined by n^2 partial derivatives (the derivatives of the n first partial derivatives with respect to the n variables):

$$\nabla^2 f = H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial {x_2}^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial {x_n}^2} \end{bmatrix}$$

If the partial derivatives are continous and f is single valued then the second-order partial derivatives can be represented by a square symmetric matrix called the **Hessian matrix**

Multivariable Non linear Optimization

Example

$$\nabla f = \begin{bmatrix} 15 - 3(x_3)^2 & 6(x_2)^2 & -6x_1x_3 \end{bmatrix}$$

$$\nabla^2 f = \begin{bmatrix} 0 & 0 & -6x_3 \\ 0 & 12x_2 & 0 \\ -6x_3 & 0 & -6x_1 \end{bmatrix}$$

Multivariable Non linear Optimization

Nearly all multivariable optimization methods do the following:

1. Starting from $\boldsymbol{x}_k$ choose a **search direction** $\boldsymbol{d}_k$
2. Minimize/maximize along that direction to find a new point:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \ \boldsymbol{d}_k$$

where k is the current iteration number and α_k is a positive scalar called the **step size**.

Steepest Descent Method

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

$$\boldsymbol{d}_k = \begin{Bmatrix} + \\ - \end{Bmatrix} \nabla f(\boldsymbol{x}_k) \quad \textbf{for} \quad \begin{Bmatrix} \max \\ \min \end{Bmatrix}$$

So, $$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k \begin{Bmatrix} + \\ - \end{Bmatrix} \alpha_k \nabla f(\boldsymbol{x}_k)$$

The Step Size

- The step size, α_k, can be calculated in the following way:

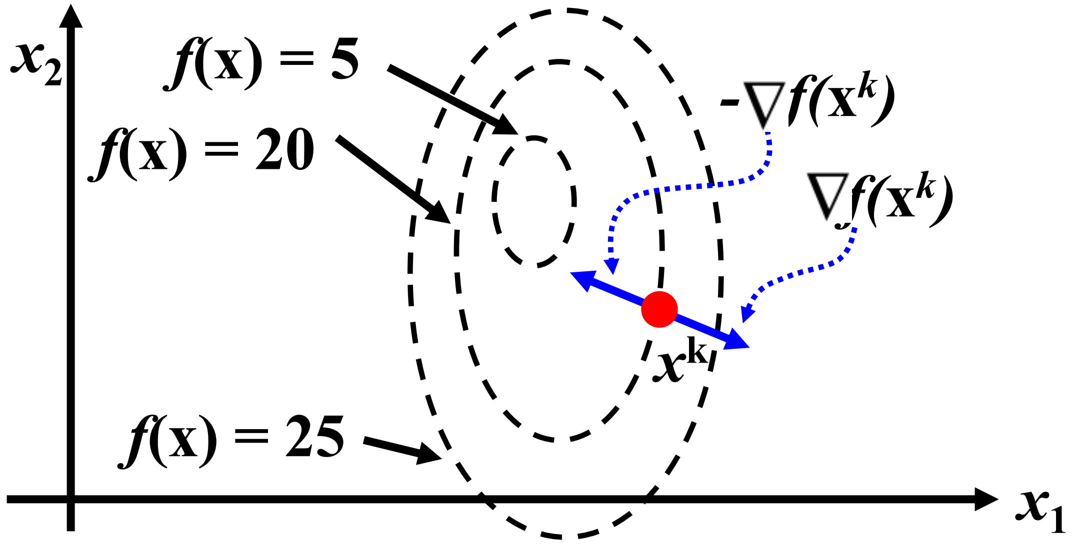
- We want to minimize/maximize the function $f(\boldsymbol{x}_{k+1}) = f(\boldsymbol{x}_k + \alpha_k \nabla f(\boldsymbol{x}_k))$ where the only variable is α_k because $\boldsymbol{x}_k$ & $\nabla f(\boldsymbol{x}_k)$ are known.

- We set $$\frac{df(\boldsymbol{x}_k + \alpha_k \nabla f(\boldsymbol{x}_k))}{d\alpha_k} = 0$$

and solve for α_k using a single-variable solution method such as the ones shown previously.

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



Steepest Descent Method Steps

So the steps of the Steepest Descent Method are:

1. Choose an initial point $\boldsymbol{x}_0$
2. Calculate the gradient $\nabla f(\boldsymbol{x}_k)$ where k is the iteration number
3. Calculate the search vector: $\boldsymbol{d}_k = \begin{Bmatrix} + \\ - \end{Bmatrix} \nabla f(\boldsymbol{x}_k)$
4. Calculate the next $\mathbf{x}_{k+1}$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k \begin{Bmatrix} + \\ - \end{Bmatrix} \alpha_k \nabla f(\boldsymbol{x}_k)$$

5. Use a single-variable optimization method to determine α_k.

Steepest Descent Method Steps

5. To determine convergence, either use some given tolerance ε_1 and evaluate:

$$|f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)| < \varepsilon_1$$

for convergence

Or, use another tolerance ε_2 and evaluate:

$$||\nabla f(\boldsymbol{x}_{k+1})|| < \varepsilon_2$$

for convergence

Convergence

- These two criteria can be used for any of the multivariable optimization methods discussed here

Recall: The norm of a vector, $\|\mathbf{x}\|$ is given by:

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \cdot \mathbf{x}} = \sqrt{(x_1)^2 + (x_2)^2 + \cdots + (x_n)^2}$$