

Minimization Using Descent Information

- we will consider the minimization of unconstrained functions of several variables where we now assume we have some derivative information such as the gradient vector or the Hessian matrix.
- Recall that Powell's method used the powerful concept of conjugate directions and performed a series of line searches.
- We will see how these conjugate directions are related to the gradient directions and we will introduce a very powerful method called the *conjugate-gradient* technique.
- Recall Taylor's expansion uses such information:

$$f(\mathbf{x} + \Delta\mathbf{x}) \cong f(\mathbf{x}) + \Delta\mathbf{x}^T \nabla f(\mathbf{x}) + \frac{1}{2} \Delta\mathbf{x}^T H(\mathbf{x}) \Delta\mathbf{x}$$

- Methods using only first derivatives are called *first-order methods*
- Methods using second order derivatives are called *second-order methods*.

The Gradient Vector Re-examined

- Recall that the gradient vector of a function $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$:

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

- Consider a differential length

$$d\mathbf{x} = \begin{bmatrix} dx_1 \\ dx_2 \\ \dots \\ dx_n \end{bmatrix} = \mathbf{u} ds$$

where $\|\mathbf{u}\| = 1$ and \mathbf{u} holds the directional information of the differential.

- Now a change in the function $f(\mathbf{x})$ along $d\mathbf{x}$ is given by

$$df = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i = (\nabla f(\mathbf{x}), d\mathbf{x}) = (\nabla f(\mathbf{x}), \mathbf{u} ds) = ds(\nabla f(\mathbf{x}), \mathbf{u})$$

the rate of change of $f(\mathbf{x})$ along the arbitrary direction \mathbf{u} is given by

$$\frac{df}{ds} = (\nabla f(\mathbf{x}), \mathbf{u})$$

- On the other hand, along a direction \mathbf{u} , the function $f(\mathbf{x})$ is described as $f(\mathbf{x} + \alpha\mathbf{u})$ and thus the rate of change along this direction is also written as

$$\frac{df}{ds} = (\nabla f(\mathbf{x}), \mathbf{u}) = \frac{d}{d\alpha} f(\mathbf{x} + \alpha\mathbf{u}) \quad (1)$$

- We can examine (1) to see which direction gives the maximum rate of increase.
- A well known inequality in functional analysis is called the *Cauchy-Schwarz* inequality which says

$$(a, b) \leq \|a\| \|b\| \quad (2)$$

Applying this to (1) we find

$$(\nabla f(\mathbf{x}), \mathbf{u}) \leq \|\nabla f(\mathbf{x})\| \|\mathbf{u}\| = \|\nabla f(\mathbf{x})\| \quad (3)$$

and letting $\mathbf{u} = \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}$ we have

$$\left(\nabla f(\mathbf{x}), \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} \right) = \frac{\|\nabla f(\mathbf{x})\|^2}{\|\nabla f(\mathbf{x})\|} = \|\nabla f(\mathbf{x})\| \quad (4)$$

which shows that for this choice of direction, the directional derivative

$$(\nabla f(\mathbf{x}), \mathbf{u})$$

reaches its maximum value.

- Therefore we say that $\nabla f(\mathbf{x})$ is the **direction of maximum increase** and
- $-\nabla f(\mathbf{x})$ is the direction of maximum decrease (also: **steepest ascent and steepest descent**).

Cauchy's Method (Steepest Descent)

- A logical minimization strategy is to use the direction of steepest descent and perform a line search in that direction.
- Assume we are at a point \mathbf{x}_k and that we have calculated the gradient at this point, $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k) = \nabla f(\mathbf{x}_k)$.
- Then we can minimize along \mathbf{g}_k starting from \mathbf{x}_k :

$$\alpha_k = \arg \min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{g}_k) \quad (5)$$

and arrive at the new point

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{g}_k \quad (6)$$

Algorithm: Cauchy's Method of Steepest Descent

1. input: $f(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$, \mathbf{x}_0 , g_{tol} , k_{max}
 2. set: $\mathbf{x} = \mathbf{x}_0$, $\mathbf{g} = \mathbf{g}(\mathbf{x})$
 3. while $k < k_{max}$ and $\|\mathbf{g}\| > g_{tol}$
 4. set: $\alpha = \arg \min_{\alpha} f(\mathbf{x} + \alpha \mathbf{g})$
 5. set: $\mathbf{x} = \mathbf{x} + \alpha \mathbf{g}$
 6. set: $\mathbf{g} = \mathbf{g}(\mathbf{x})$, $k = k + 1$
 7. end
- Note that we don't bother to take the negative of \mathbf{g} in the line search since this is automatically accomplished by allowing negative values of α .
 - The iterations end when either a maximum number of iterations have been reached or the norm of the gradient at the current point is less than a user defined value g_{tol} which is a value close to zero.

- The successive directions of minimization in the method of steepest descent are orthogonal to each other.
- This can be shown as follows: assume that we are at a point \mathbf{x}_k and we need to find

$$\alpha_k = \arg \min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{g}_k) \quad (7)$$

in order to arrive at the new point

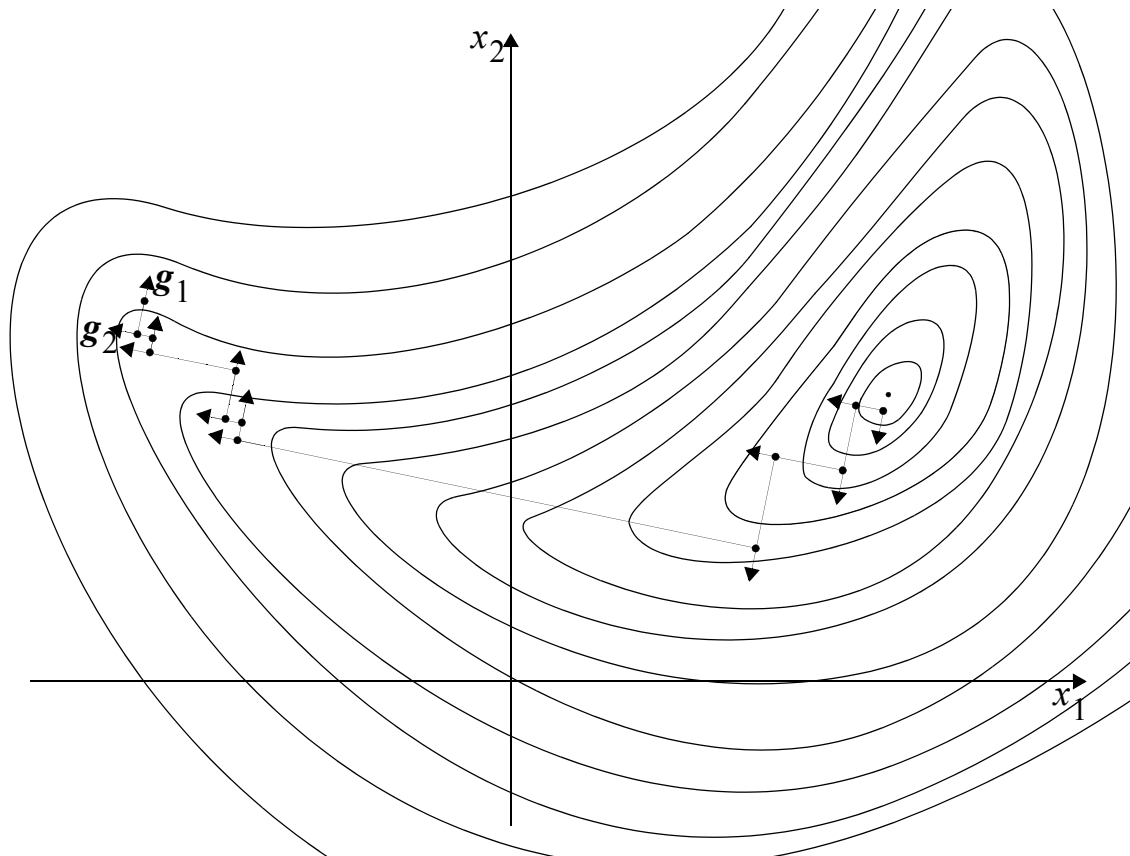
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{g}_k. \quad (8)$$

- We can find α_k by setting the derivative of $F(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{g}_k)$ with respect to α equal to zero, which from (1) we can write:

$$\frac{d}{d\alpha} F(\alpha) = (\nabla f(\mathbf{x}_{k+1}), \mathbf{g}_k) = (\mathbf{g}_{k+1}, \mathbf{g}_k) = 0 \quad . \quad (9)$$

this immediately shows the orthogonality between the successive directions \mathbf{g}_{k+1} and \mathbf{g}_k .

- Now, although it may seem like a good idea to minimize along the direction of steepest descent, it turns out that this method is not very efficient.
- For relatively complicated functions, this method will tend to zig-zag towards the minimum.



Zig-zaging effect of Cauchy's method.

- Note that since successive directions of descent are always orthogonal, in two dimensions the algorithm searches in only two directions.
- This is what makes the algorithm slow to converge to the minimum and is generally not recommended.

The Conjugate-Gradient Method

- It turns out that if we have access to the gradient of the objective function, then we can determine conjugate directions relatively efficiently.
- Recall that Powell's conjugate direction method requires n single variable minimizations per iteration in order to determine one new conjugate direction at the end of the iteration.

This results in approximately n^2 line minimizations to find the minimum of a quadratic function.

- In the conjugate-gradient algorithm, access to the gradient of $f(\mathbf{x})$, that is $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})$, allows us to set up a new conjugate direction after every line minimization.
- Consider the quadratic function

$$Q(\mathbf{x}) = a + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T C \mathbf{x} \quad (10)$$

where $\mathbf{x} \in \mathbb{R}^n$.

- We want to perform successive line minimizations along conjugate directions, say $\mathbf{s}(\mathbf{x}_k)$, where \mathbf{x}_k is the current search point and we minimize to the next point as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{s}(\mathbf{x}_k). \quad (11)$$

- Now, how do we find these conjugate directions

$$\mathbf{s}_k = \mathbf{s}(\mathbf{x}_k)$$

given previous information?

- Expand the search directions in terms of the gradient at the current point

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k)$$

and a linear combination of the previous search directions:

$$\mathbf{s}_k = -\mathbf{g}_k + \sum_{i=0}^{k-1} \gamma_i \mathbf{s}_i \quad (12)$$

where we start with the initial search direction as the steepest descent direction

$$\mathbf{s}_0 = -\mathbf{g}_0.$$

- The coefficients of the expansion, γ_i , $i = 1 \dots k-1$, are to be chosen so that the \mathbf{s}_k are **C-conjugate**.

- Therefore, we have

$$\mathbf{s}_1 = -\mathbf{g}_1 + \gamma_0 \mathbf{s}_0 = -\mathbf{g}_1 - \gamma_0 \mathbf{g}_0 \quad (13)$$

and we require that

$$(\mathbf{s}_1, C\mathbf{s}_0) = 0 = (-\mathbf{g}_1 - \gamma_0 \mathbf{g}_0, C\mathbf{s}_0) \quad (14)$$

but $\mathbf{x}_1 = \mathbf{x}_0 + \lambda_0 \mathbf{s}_0$ which can be solved for \mathbf{s}_0 as

$$\frac{\mathbf{x}_1 - \mathbf{x}_0}{\lambda_0} = \mathbf{s}_0 = \frac{\Delta \mathbf{x}_0}{\lambda_0} \quad (15)$$

where $\Delta \mathbf{x}_0 = \mathbf{x}_1 - \mathbf{x}_0$ is the forward difference operator. Using this in (14) we have

$$\left(-\mathbf{g}_1 - \gamma_0 \mathbf{g}_0, C \frac{\Delta \mathbf{x}_0}{\lambda_0} \right) = 0 \quad (16)$$

but $\mathbf{g}(\mathbf{x}) = C\mathbf{x} + \mathbf{b}$ which means that we can set

$$\Delta \mathbf{g}_0 = \mathbf{g}_1 - \mathbf{g}_0 = C(\mathbf{x}_1 - \mathbf{x}_0) = C\Delta \mathbf{x}_0 \quad (17)$$

which substituting into (16), we have

$$\left(-\mathbf{g}_1 - \gamma_0 \mathbf{g}_0, \frac{\Delta \mathbf{g}_0}{\lambda_0} \right) = 0$$

$$(-\mathbf{g}_1, \Delta \mathbf{g}_0) = \gamma_0 (\mathbf{g}_0, \Delta \mathbf{g}_0)$$

and finally

$$\gamma_0 = -\frac{(\Delta \mathbf{g}_0, \mathbf{g}_1)}{(\Delta \mathbf{g}_0, \mathbf{g}_0)}. \quad (18)$$

- We can further reduce the numerator as follows:

$$(\Delta \mathbf{g}_0, \mathbf{g}_1) = (\mathbf{g}_1 - \mathbf{g}_0, \mathbf{g}_1) = (\mathbf{g}_1, \mathbf{g}_1) - (\mathbf{g}_0, \mathbf{g}_1) = \|\mathbf{g}_1\|^2 \quad (19)$$

since $(\mathbf{g}_0, \mathbf{g}_1) = 0$ (successive directions of minimization are orthogonal)

- The denominator can also be rewritten as

$$(\Delta \mathbf{g}_0, \mathbf{g}_0) = (\mathbf{g}_1 - \mathbf{g}_0, \mathbf{g}_0) = (\mathbf{g}_1, \mathbf{g}_0) - (\mathbf{g}_0, \mathbf{g}_0) = -\|\mathbf{g}_0\|^2 \quad (20)$$

and therefore the coefficient γ_0 can be calculated from

$$\gamma_0 = \frac{\|\mathbf{g}_1\|^2}{\|\mathbf{g}_0\|^2}. \quad (21)$$

- The conjugate direction, \mathbf{s}_1 , can be calculated as

$$\mathbf{s}_1 = -\mathbf{g}_1 + \frac{\|\mathbf{g}_1\|^2}{\|\mathbf{g}_0\|^2} \mathbf{s}_0. \quad (22)$$

- Now although we derived this for s_1 , we could have derived it for any s_k to get

$$\boxed{s_k = -\mathbf{g}_k + \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2} s_{k-1}} \quad (23)$$

- This represents the *Fletcher-Reeves* scheme (1964) for determining conjugate directions for search from the gradient at the current point, \mathbf{g}_k , the previous search direction, s_{k-1} , and the magnitude of the gradient at the previous point, $\|\mathbf{g}_{k-1}\|$.

- Two alternative update equations are the *Hestenes-Stiefel* method (1952):

$$\boxed{s_k = -\mathbf{g}_k + \frac{(\Delta \mathbf{g}_{k-1}, \mathbf{g}_k)}{(\Delta \mathbf{g}_{k-1}, s_k)} s_{k-1}} \quad (24)$$

and the *Polak-Ribière* method (1969):

$$\boxed{s_k = -\mathbf{g}_k + \frac{(\Delta \mathbf{g}_{k-1}, \mathbf{g}_k)}{\|\mathbf{g}_{k-1}\|^2} s_{k-1}} \quad (25)$$

- All three schemes are identical for quadratic functions but will be different for non-quadratic functions.
- For quadratic functions, these methods will find the exact minimum in n iterations.
- For non-quadratic functions, the search direction is reset to the steepest descent direction every n iterations.

Algorithm: Conjugate Gradient Method (*Fletcher-Reeves*)

1. input: $f(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$, \mathbf{x}_0 , g_{tol} , k_{max} , n
2. set: $k = 0$, $\mathbf{x} = \mathbf{x}_0$
3. set: $\mathbf{g}_{old} = \mathbf{g}_{new} = \mathbf{g}(\mathbf{x})$
4. set: $\mathbf{s}_{old} = \mathbf{s}_{new} = \mathbf{g}_{old}$
5. while $\|\mathbf{g}_{new}\| > g_{tol}$ and $k < k_{max}$
6. $k = k + 1$
7. for $i = 1(1)n$
8. if $i = 1$ then
9. set: $\mathbf{s}_{new} = -\mathbf{g}_{new}$
10. else
11. set: $\gamma = \frac{\|\mathbf{g}_{new}\|^2}{\|\mathbf{g}_{old}\|^2}$
12. set: $\mathbf{s}_{new} = -\mathbf{g}_{new} + \gamma\mathbf{s}_{old}$
13. end
14. set: $\lambda = \arg \min_{\lambda} f(\mathbf{x} + \lambda\mathbf{s}_{new})$
15. set: $\mathbf{x} = \mathbf{x} + \lambda\mathbf{s}_{new}$
16. set: $\mathbf{g}_{old} = \mathbf{g}_{new}$, $\mathbf{g}_{new} = \mathbf{g}(\mathbf{x})$
17. set: $\mathbf{s}_{old} = \mathbf{s}_{new}$
18. end
19. end

- Notes: line 11 contains the *Fletcher-Reeves* update scheme.
- This can be changed to use the *Hestenes-Stiefel* or the *Polak-Ribière* as desired (*i.e.* (24) or (25) respectively); the algorithm terminates once the norm of the gradient is smaller than a user specified value g_{tol} or the number of iterations grow beyond k_{max} .