Optimization

Hassan OMRAN

Lecture 3: Multi-Dimensional Search Methods - part II

Télécom Physique Strasbourg
Université de Strasbourg

Outline of the talk

1. Conjugate direction methods

2. Quasi-Newton methods

1. Conjugate direction methods

2. Quasi-Newton methods

Conjugate direction methods

This method does not requires inverting a matrix. Also, it can be implemented without the calculation of the Hessian. It is based on the notion of Q-conjugate directions.

Definition 1

For a symmetric matrix $Q = Q^T \in \mathbb{R}^{n\times n}$, the directions $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_m$ are called **Q-conjugate** if

$$\mathbf{d}_i^T Q \mathbf{d}_j = 0, \qquad \forall i \neq j \tag{1}$$

When $Q \succ 0$:

Theorem 2

Let $Q = Q^T \in \mathbb{R}^{n\times n}$ such that $Q \succ 0$. If $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_k$, $k \leq n-1$ are nonzero Q-conjugate, then they are linearly independent.

Conjugate direction methods

Proof.

Consider $\alpha_0, \alpha_1, \ldots, \alpha_k$ *such that*

$$\alpha_0 \boldsymbol{d}_0 + \alpha_1 \boldsymbol{d}_1 + \cdots + \alpha_k \boldsymbol{d}_k = \boldsymbol{0}$$

multiplying by $\boldsymbol{d}_j^T Q$ *for* $0 \leq j \leq k$

$$\alpha_j \boldsymbol{d}_j^T Q \boldsymbol{d}_j = 0$$

Since $Q \succ 0$ *and* $\boldsymbol{d}_j \neq \boldsymbol{0}$, *then* $\alpha_j = 0$ *for* $0 \leq j \leq k$ □

Remark 1.1

Note that for $Q^T = Q \succ 0$, *then* n *nonzero* Q*-conjugate directions define a basis for* $\mathbb{R}^n$.

Conjugate direction methods: quadratic functions

The case of quadratic function

Consider the following problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{q}^T \mathbf{x} \\ s.t. \quad & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{2}$$

for a matrix $0 \prec Q = Q^T \in \mathbb{R}^{n \times n}$ *and a vector* $\mathbf{q} \in \mathbb{R}^n$. *Note that* $\nabla f(\mathbf{x}) = Q\mathbf{x} + \mathbf{q}$ *and* $D^2 f(\mathbf{x}^\star) = Q \succ 0$.

Given the initial point $\mathbf{x}_0$, *and Q-conjugate directions* $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}$, *the idea is to perform at iteration k a one-dimensional optimization according to the direction* $\mathbf{d}_k$ *and start the next iteration at the found minimizer*

That is, at each iteration we have

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad \text{with} \quad \alpha_k = \arg\min_{\alpha \geq 0} f\left(\mathbf{x}_k + \alpha \mathbf{d}_k\right)$$

Conjugate direction methods: quadratic functions

Consider the function $h_k(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$, *then*

$$0 = \dot{h}_k(\alpha)\big|_{\alpha=\alpha_k} = \left(\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)\right)^T \mathbf{d}_k = \left(Q(\mathbf{x}_k + \alpha_k \mathbf{d}_k) + \mathbf{q}\right)^T \mathbf{d}_k \tag{3}$$

$$\Rightarrow \alpha_k = -\frac{(Q\mathbf{x}_k + \mathbf{q})^T \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k} = -\frac{\nabla f(\mathbf{x}_k)^T \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k} \tag{4}$$

Note that from (3) *we have also proved that*

$$0 = \dot{h}_k(\alpha)\big|_{\alpha=\alpha_k} = \left(\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)\right)^T \mathbf{d}_k = \left(\nabla f(\mathbf{x}_{k+1})\right)^T \mathbf{d}_k$$

thus

$$\nabla f(\mathbf{x}_{k+1})^T \mathbf{d}_k = 0, \qquad \forall k \in \{0, \cdots, n-1\} \tag{5}$$

Conjugate direction methods: quadratic functions

For simplicity, we will use the following notation $\boldsymbol{g}_k := \nabla f(\boldsymbol{x}_k)$

Basic Conjugate Direction Algorithm:
with any initial condition $\boldsymbol{x}_0$ *and and* Q*-conjugate directions* $\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n-1}$

$$\boldsymbol{g}_k = Q\boldsymbol{x}_k + \boldsymbol{q} \tag{6}$$

$$\alpha_k = -\frac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{\boldsymbol{d}_k^T Q \boldsymbol{d}_k} \tag{7}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k \tag{8}$$

Conjugate direction methods: quadratic functions

Note that in (5) *it has been proved that*

$$\boldsymbol{g}_{k+1}^T \boldsymbol{d}_k = 0, \qquad \forall k \in \{0, \cdots, n-1\}$$

In fact, the last property is valid also for nonquadratic functions. For the case of quadratic functions, the algorithm has even the following stronger property

Theorem 3

Consider the problem in (2)*. The conjugate directions algorithm has the following property*

$$\boldsymbol{g}_{k+1}^T \boldsymbol{d}_i = 0, \qquad \forall i \in \{0, \ldots, k\}, \quad \forall k \in \{0, \ldots, n-1\} \tag{9}$$

That is the gradient at iteration $k+1$ *is orthogonal to all directions from previous iterations*

$$\begin{aligned}
&\boldsymbol{g}_1^T \boldsymbol{d}_0 = 0, \\
&\boldsymbol{g}_2^T \boldsymbol{d}_0 = 0, \ \boldsymbol{g}_2^T \boldsymbol{d}_1 = 0, \\
&\vdots \qquad\qquad\qquad\qquad \ddots \\
&\boldsymbol{g}_n^T \boldsymbol{d}_0 = 0, \ \boldsymbol{g}_n^T \boldsymbol{d}_1 = 0, \cdots, \boldsymbol{g}_n^T \boldsymbol{d}_{n-1} = 0.
\end{aligned}$$

Conjugate direction methods: quadratic functions

Proof

*We proceed by induction on k. First, for $k = 0$ we have $\boldsymbol{g}_1^T \boldsymbol{d}_0 = 0$ from (5).
Suppose the result holds for k, that is*

$$\boldsymbol{g}_k^T \boldsymbol{d}_0 = 0, \cdots, \boldsymbol{g}_k^T \boldsymbol{d}_{k-1} = 0, \tag{10}$$

and we will proof the result for $k+1$, that is

$$\boldsymbol{g}_{k+1}^T \boldsymbol{d}_0 = 0, \cdots, \boldsymbol{g}_{k+1}^T \boldsymbol{d}_{k-1} = 0, \boldsymbol{g}_{k+1}^T \boldsymbol{d}_k = 0 \tag{11}$$

First note that

$$\boldsymbol{g}_{k+1} - \boldsymbol{g}_k = (Q\boldsymbol{x}_{k+1} + \boldsymbol{q}) - (Q\boldsymbol{x}_k + \boldsymbol{q}) = Q(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) = \alpha_k Q\boldsymbol{d}_k$$

thus

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

Conjugate direction methods: quadratic functions

Proof (Cont.)

by taking the inner product of the two sides of the previous equality by $\boldsymbol{d}_i$ for $i \in \{0, \dots, k-1\}$

$$\boldsymbol{g}_{k+1}^T \boldsymbol{d}_i = \boldsymbol{g}_k^T \boldsymbol{d}_i + \alpha_k \boldsymbol{d}_k^T Q \boldsymbol{d}_i = 0, \qquad i \in \{0, \dots, k-1\} \tag{12}$$

where the last equality is from (10) and from $\boldsymbol{d}_k^T Q \boldsymbol{d}_i = 0$ by Q-conjugacy.
Finally (12) is also satisfied for $i = k$ from (5). This proves that

$$\boldsymbol{g}_{k+1}^T \boldsymbol{d}_i = 0, \qquad i \in \{0, \dots, k\}$$

Which proofs (11).

□

This shows that the conjugate direction method algorithm converges in n steps (for quadratic functions). This can be seen from $\boldsymbol{g}_n^T \boldsymbol{d}_i = 0 \; \forall i \in \{0, \dots, n-1\}$ which means that $\boldsymbol{g}_n$ is orthogonal to a space spanned by $\{\boldsymbol{d}_0, \cdots, \boldsymbol{d}_{n-1}\} = \mathbb{R}^n \Rightarrow Q\boldsymbol{x}_n + \boldsymbol{q} = \boldsymbol{g}_n = \boldsymbol{0}$, thus $\boldsymbol{x}^\star = \boldsymbol{x}_n$.

In the following another proof is presented.

Conjugate direction methods: quadratic functions

Theorem 4

Consider the problem in (2)*. Then, the conjugate direction algorithm converges the solution* $\boldsymbol{x}^\star = -Q^{-1}\boldsymbol{q}$ *in* n *iterations* $\forall \boldsymbol{x}_0 \in \mathbb{R}^n$*.*

Proof

By remark 1.1 there exist n *scalars* $\beta_0, \dots, \beta_{n-1}$ *such that*

$$\boldsymbol{x}^\star - \boldsymbol{x}_0 = \sum_{i=0}^{n-1} \beta_i \boldsymbol{d}_i \tag{13}$$

Also, from (8) *we have*

$$\boldsymbol{x}_n = \boldsymbol{x}_0 + \alpha_0 \boldsymbol{d}_0 + \alpha_1 \boldsymbol{d}_1 + \cdots + \alpha_{n-1} \boldsymbol{d}_{n-1}$$

$$\boldsymbol{x}_n - \boldsymbol{x}_0 = \sum_{i=0}^{n-1} \alpha_i \boldsymbol{d}_i \tag{14}$$

Conjugate direction methods: quadratic functions

Proof (Cont.)

Subtracting (13) *from* (14) *we get*

$$(\mathbf{x}_n - \mathbf{x}^\star) = \sum_{i=0}^{n-1} (\alpha_i - \beta_i)\mathbf{d}_i \tag{15}$$

Premultiplying both sides by $\mathbf{d}_k^T Q$ $\forall k \in \{0, \dots, n-1\}$

$$\begin{aligned} \mathbf{d}_k^T \underbrace{Q(\mathbf{x}_n - \mathbf{x}^\star)}_{=Q\mathbf{x}_n + \mathbf{q} = \mathbf{g}_n} &= \sum_{i=0}^{n-1} (\alpha_i - \beta_i)\mathbf{d}_k^T Q \mathbf{d}_i, \qquad \forall k \in \{0, \dots, n-1\} \\ 0 = \mathbf{d}_k^T \mathbf{g}_n &= (\alpha_k - \beta_k)\mathbf{d}_k^T Q \mathbf{d}_k, \qquad \forall k \in \{0, \dots, n-1\} \end{aligned}$$

where the left equality is from Theorem 3, and since $\mathbf{d}_k^T Q \mathbf{d}_k > 0$ *(Q is positive definite) then* $\alpha_k = \beta_k$ $\forall k \in \{0, \dots, n-1\}$.
Finally, since $\alpha_k = \beta_k$ *we have from* (15) $\mathbf{x}^\star = \mathbf{x}_n$

□

Conjugate direction methods: generating the directions

Till now we supposed that there exist n Q-conjugate directions. Here we examine a method which permits to generate these directions.

The following method is based on the the Gram-Schmidt process

Given an arbitrary set of linear independent vectors $\{\boldsymbol{p}_0, \cdots, \boldsymbol{p}_{n-1}\}$, *generate the vectors* $\{\boldsymbol{d}_0, \cdots, \boldsymbol{d}_{n-1}\}$:

$$\boldsymbol{d}_0 = \boldsymbol{p}_0 \tag{16}$$

$$\boldsymbol{d}_{k+1} = \boldsymbol{p}_{k+1} - \sum_{i=0}^{k} \frac{\boldsymbol{p}_{k+1}^T Q \boldsymbol{d}_i}{\boldsymbol{d}_i^T Q \boldsymbol{d}_i} \boldsymbol{d}_i \tag{17}$$

<u>Exercise:</u> show that the directions generated using (16) (17) *are Q-conjugate.*

Conjugate direction methods: generating the directions

Solution: *This can be proved by induction. First, note that*

$$\boldsymbol{d}_1 = \boldsymbol{p}_1 - \frac{\boldsymbol{p}_1^T Q \boldsymbol{d}_0}{\boldsymbol{d}_0^T Q \boldsymbol{d}_0} \boldsymbol{d}_0$$

thus $\boldsymbol{d}_1$ *is a linear combination of* $\boldsymbol{p}_0$ *and* $\boldsymbol{p}_1$ *and*

$$\boldsymbol{d}_0^T Q \boldsymbol{d}_1 = \boldsymbol{d}_0^T Q \boldsymbol{p}_1 - \frac{\boldsymbol{p}_1^T Q \boldsymbol{d}_0}{\boldsymbol{d}_0^T Q \boldsymbol{d}_0} \boldsymbol{d}_0^T Q \boldsymbol{d}_0 = 0$$

Now suppose that $\boldsymbol{d}_j^T Q \boldsymbol{d}_i = 0 \; \forall i \neq j \in \{1, \cdots, k\}$*, and that* $\boldsymbol{d}_k$ *is a liner combination of* $\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$*. First, from* (17) *we see that* $\boldsymbol{d}_{k+1}$ *is a linear combination of* $\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{k+1}$ *who are linear independent (thus* $\boldsymbol{d}_{k+1} \neq \boldsymbol{0}$*). Moreover*

$$\begin{aligned}
\boldsymbol{d}_j^T Q \boldsymbol{d}_{k+1} &= \boldsymbol{d}_j^T Q \boldsymbol{p}_{k+1} - \sum_{i=0}^{k} \frac{\boldsymbol{p}_{k+1}^T Q \boldsymbol{d}_i}{\boldsymbol{d}_i^T Q \boldsymbol{d}_i} \underbrace{\boldsymbol{d}_j^T Q \boldsymbol{d}_i}_{=0 \text{ for } i \neq j} \qquad \forall j \in \{0, \cdots, k\} \\
&= \boldsymbol{d}_j^T Q \boldsymbol{p}_{k+1} - \frac{\boldsymbol{p}_{k+1}^T Q \boldsymbol{d}_j}{\boldsymbol{d}_j^T Q \boldsymbol{d}_j} \boldsymbol{d}_j^T Q \boldsymbol{d}_j \\
&= 0
\end{aligned}$$

Conjugate gradient algorithm

*We have seen that it is possible to generate the Q-conjugate directions before starting the iterations. This however can be avoided. The **conjugate gradient algorithm** generates a new Q-conjugate direction at each iteration.*

Conjugate Gradient Algorithm:

with any initial condition $\mathbf{x}_0$ *and* $\mathbf{d}_0 = -\mathbf{g}_0$*:*

$$\alpha_k = -\frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k} \tag{18}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \tag{19}$$

$$\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}) = Q\mathbf{x}_{k+1} + \mathbf{q} \tag{20}$$

$$\beta_k = \frac{\mathbf{g}_{k+1}^T Q \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k} \tag{21}$$

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k \tag{22}$$

And if at any iteration $\mathbf{g}_k = \nabla f(\mathbf{x}_k) = \mathbf{0}$ *then stop*

Conjugate gradient algorithm

Theorem 5

The directions $\{\boldsymbol{d}_0, \cdots, \boldsymbol{d}_{n-1}\}$ in the conjugate gradient algorithm are Q-conjugate, and

$$\boldsymbol{g}_{k+1}^T \boldsymbol{g}_j = 0, \qquad \forall j \in \{0, \cdots, k\}, \qquad \forall k \in \{0, \cdots, n-1\} \tag{23}$$

Proof

We proceed by induction. First, note that

$$\begin{aligned} \boldsymbol{d}_0^T Q \boldsymbol{d}_1 &= \boldsymbol{d}_0^T Q(-\boldsymbol{g}_1 + \beta_0 \boldsymbol{d}_0) \\ &= \boldsymbol{d}_0^T Q(-\boldsymbol{g}_1 + \frac{\boldsymbol{g}_1^T Q \boldsymbol{d}_0}{\boldsymbol{d}_0^T Q \boldsymbol{d}_0} \boldsymbol{d}_0) = 0 \end{aligned}$$

Also, by Theorem 3

$$\boldsymbol{g}_1^T \boldsymbol{g}_0 = -\boldsymbol{g}_1^T \boldsymbol{d}_0 = 0$$

Now suppose that $\{\boldsymbol{d}_0, \cdots, \boldsymbol{d}_k\}$ are Q-conjugated, and let us prove the case for $k+1$.

Conjugate gradient algorithm

Proof (Cont.)

First, from Theorem 3

$$\boldsymbol{g}_{k+1}^T \boldsymbol{d}_j = 0, \quad j \in \{0, \ldots, k\}$$

This shows that ($j = 0$)

$$\boldsymbol{g}_{k+1}^T \boldsymbol{g}_0 = -\boldsymbol{g}_{k+1}^T \boldsymbol{d}_0 = 0 \tag{24}$$

and

$$\boldsymbol{g}_{k+1}^T \boldsymbol{g}_j = \boldsymbol{g}_{k+1}^T(-\boldsymbol{d}_j + \beta_{j-1}\boldsymbol{d}_{j-1}) = 0, \quad j \in \{1, \ldots, k\} \tag{25}$$

From (24) and (25), we have that

$$\boldsymbol{g}_{k+1}^T \boldsymbol{g}_j = 0, \qquad \forall j \in \{0, \ldots, k\} \tag{26}$$

Now we consider $\boldsymbol{d}_{k+1}^T Q \boldsymbol{d}_j$ for $j \in \{0, \ldots, k\}$. First, for $j \in \{0, \ldots, k-1\}$

$$\begin{aligned} \boldsymbol{d}_{k+1}^T Q \boldsymbol{d}_j &= (-\boldsymbol{g}_{k+1} + \beta_k \boldsymbol{d}_k)^T Q \boldsymbol{d}_j \\ &= -\boldsymbol{g}_{k+1}^T Q \boldsymbol{d}_j \end{aligned} \tag{27}$$

Conjugate gradient algorithm

Proof (Cont.)

and since $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{d}_j$ *then by multiplying by* Q *from the left and adding* $\mathbf{q}$ *we get*

$$\begin{aligned} Q\mathbf{x}_{j+1} + \mathbf{q} &= Q\mathbf{x}_j + \mathbf{q} + \alpha_j Q\mathbf{d}_j \\ \mathbf{g}_{j+1} &= \mathbf{g}_j + \alpha_j Q\mathbf{d}_j \end{aligned}$$

thus by replacing the term $Q\mathbf{d}_j$ *in* (27) *we get*

$$\mathbf{d}_{k+1}^T Q\mathbf{d}_j = -\mathbf{g}_{k+1}^T\left(\frac{\mathbf{g}_{j+1} - \mathbf{g}_j}{\alpha_j}\right) = 0, \qquad \forall j \in \{0, \dots, k-1\} \tag{28}$$

where the last equality is from (26). *Finally, we still need to show the case* $j = k$, *that is:*

$$\mathbf{d}_{k+1}^T Q\mathbf{d}_k = (-\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k)^T Q\mathbf{d}_k = (-\mathbf{g}_{k+1} + \frac{\mathbf{g}_{k+1}^T Q\mathbf{d}_k}{\mathbf{d}_k^T Q\mathbf{d}_k}\mathbf{d}_k)^T Q\mathbf{d}_k = 0 \tag{29}$$

Thus from (29) *and* (28) *we have that* $\mathbf{d}_{k+1}^T Q\mathbf{d}_j = 0,\ \forall j \in \{0, \dots, k\}$ *which completes the proof.* □

Conjugate direction methods: the non-quadratic case

- *The method can be extended to the non quadratic case by finding a quadratic approximation of the objective function at each step*
- *Evaluating the Hessian at each step might be computationally hard, this is why we will look for a method that avoids calculating the Hessian*
- *Note that the Hessian appears in two expressions in the conjugate gradient algorithm:*

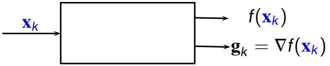
  $\rightarrow \alpha_k$ *which can be solved by a line search:* $\alpha_k = \arg\min_{\alpha \geq 0} f\left(\mathbf{x}_k + \alpha \mathbf{d}_k\right)$

  $\rightarrow \beta_k$ *for which we show next how to avoid using calculating the Hessian*

Conjugate direction methods: the non-quadratic case

The Fletcher Reeves conjugate method:

In order to find a method which avoids calculating the Hessian, consider again the case of quadratic functions $f(\mathbf{x}_k) = \frac{1}{2}\mathbf{x}_k{}^T Q\mathbf{x}_k + \mathbf{q}^T\mathbf{x}_k$. *We will find a solution for this case and generalize it.*



The block calculates the values of $f(\mathbf{x}_k)$ and the gradient.

So we suppose that we are able to get the value of the function and the value of the gradient but <u>*the Hessian is unknown*</u>*. Now the question is*

> *How can we modify the conjugate gradient to make it applicable without calculating the Hessian ?*

Conjugate direction methods: the non-quadratic case

The Fletcher Reeves conjugate method:

Note that what we are trying to do is to replace Q in the expression of $\beta_k = \frac{\mathbf{g}_{k+1}^T Q \mathbf{d}_k}{\mathbf{d}_k^T Q \mathbf{d}_k}$

First, since $\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{d}_j$ *then by multiplying by Q from the left and adding* $\mathbf{q}$ *we get*

$$\underbrace{Q\mathbf{x}_{k+1} + \mathbf{q}}_{\mathbf{g}_{k+1}} = \underbrace{Q\mathbf{x}_k + \mathbf{q}}_{\mathbf{g}_k} + \alpha_k Q \mathbf{d}_k$$

Thus,

$$Q\mathbf{d}_k = \frac{\mathbf{g}_{k+1} - \mathbf{g}_k}{\alpha_k}$$

thus by replacing the $Q\mathbf{d}_k$ *in the expression of* β_k *we get*

$$\beta_k = \frac{\mathbf{g}_{k+1}^T \left(\frac{\mathbf{g}_{k+1} - \mathbf{g}_k}{\alpha_k}\right)}{\mathbf{d}_k^T \left(\frac{\mathbf{g}_{k+1} - \mathbf{g}_k}{\alpha_k}\right)} = \frac{\overbrace{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}^{t_1} \overbrace{- \mathbf{g}_{k+1}^T \mathbf{g}_k}^{t_2}}{\underbrace{\mathbf{d}_k^T \mathbf{g}_{k+1}}_{t_3} \underbrace{- \mathbf{d}_k^T \mathbf{g}_k}_{t_4}} \tag{30}$$

Conjugate direction methods: the non-quadratic case

The Fletcher Reeves conjugate method:

Note that $t_2 = 0$ (by Theorem 5) and $t_3 = 0$ (by Theorem 3).
Finally

$$\begin{aligned} t_4 &= -\boldsymbol{d}_k^T \boldsymbol{g}_k \\ &= -(-\boldsymbol{g}_k + \beta_{k-1}\boldsymbol{d}_{k-1})^T \boldsymbol{g}_k \\ &= \boldsymbol{g}_k^T \boldsymbol{g}_k \end{aligned}$$

which shows that

$$\beta_k = \frac{\boldsymbol{g}_{k+1}^T \boldsymbol{g}_{k+1}}{\boldsymbol{g}_k^T \boldsymbol{g}_k} \tag{31}$$

*which defines the **Fletcher Reeves conjugate formula**.*

*There are other formulas for applying congregate methods to nonlinear functions such as **Hestenes-Stiefel** $\beta_k = \frac{\boldsymbol{g}_{k+1}^T(\boldsymbol{g}_{k+1} - \boldsymbol{g}_k)}{\boldsymbol{d}_k^T(\boldsymbol{g}_{k+1} - \boldsymbol{g}_k)}$ and **Polak-Ribière** $\beta_k = \frac{\boldsymbol{g}_{k+1}^T(\boldsymbol{g}_{k+1} - \boldsymbol{g}_k)}{\boldsymbol{g}_k^T \boldsymbol{g}_k}$.*

Conjugate direction methods: the non-quadratic case

The Fletcher Reeves conjugate method:

with any initial condition $\mathbf{x}_0$ *and* $\mathbf{d}_0 = -\mathbf{g}_0$*:*

$$\alpha_k = \arg\min_{\alpha \geq 0} f\left(\mathbf{x}_k + \alpha \mathbf{d}_k\right)$$

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

$$\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$$

$$\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}$$

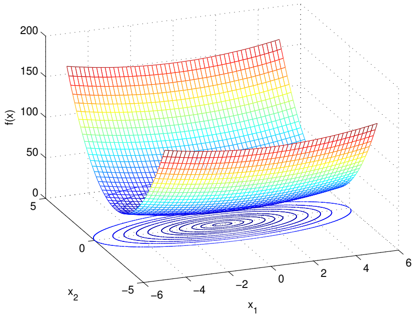
$$\mathbf{d}_{k+1} = -\mathbf{g}^{k+1} + \beta_k \mathbf{d}_k$$

And if at any iteration $\mathbf{g}^k = \nabla f(\mathbf{x}_k) = \mathbf{0}$ *then stop*

Conjugate direction methods: comparison with gradient methods

Consider the following quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \boldsymbol{x} + [0 \;\; 0]\boldsymbol{x} + cte$$



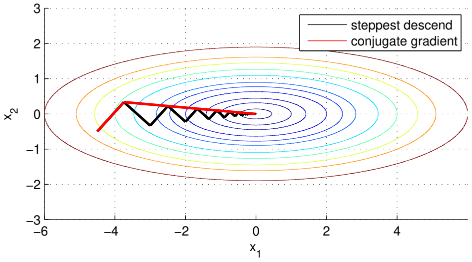
The level sets of the considered quadratic function

Conjugate direction methods: comparison with gradient methods

Consider the following quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \boldsymbol{x} + [0 \;\; 0]\boldsymbol{x} + cte$$

The next figure shows the sequences resulting from the steepest descent and conjugate directions methods



Conjugate direction methods: remarks

- *This method can be seen as an intermediate method between the steepest descent and Newton's method*
- *For a quadratic function with n variables, the method converges in n steps*
- *No matrix storage is needed*
- *Note that the accuracy of the line search has a great influence on the performance of this method*
- *For nonquadratic functions, the algorithm will not converge in n steps, and practical issues should be considered:*

  → *A stopping criteria should be considered instead of $\nabla f(\mathbf{x}_k) = 0$*

  → *The choice of the formula for β_k depends on the objective function*

  → *The Q-conjugacy of the generated directions might deteriorate. A practical solution is to reinitialize the direction vector to $-\nabla f(\mathbf{x}_k)$ each few iterations*

1. Conjugate direction methods

2. Quasi-Newton methods

Quasi-Newton methods

Newton's method is regarded as one of the most successful methods for optimization, but it has some computational drawbacks: it requires the calculation of the Hessian, and solving a set of linear equations.

> *The idea of quasi-Newton methods is to construct approximations of the inverse of the Hessian matrix, thus there will be no need for the calculation of the Hessian nor the solution a set of linear equations.*

That is, instead of Newton's method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \left(D^2 f(\mathbf{x}_k)\right)^{-1} \nabla f(\mathbf{x}_k), \quad \text{with } \alpha_k = \arg\min_{\alpha \geq 0} f\left(\mathbf{x}_k - \alpha \left(D^2 f(\mathbf{x}_k)\right)^{-1} \nabla f(\mathbf{x}_k)\right)$$

we consider

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k \nabla f(\mathbf{x}_k), \quad \text{with } \alpha_k = \arg\min_{\alpha \geq 0} f\left(\mathbf{x}_k - \alpha \mathbf{H}_k \nabla f(\mathbf{x}_k)\right)$$

Where $\mathbf{H}_0, \mathbf{H}_1, \ldots$ are estimates of the inverse of the Hessian $D^2 f(\mathbf{x}_k)$. Note that approximating the second derivative is the basis for the secant method for the case of one-dimensional functions.

Quasi-Newton methods: conditions on $\mathbf{H}_k$

Consider the case of quadratic functions $f(\boldsymbol{x}_k) = \frac{1}{2}\boldsymbol{x}_k{}^T Q \boldsymbol{x}_k + \boldsymbol{q}^T \boldsymbol{x}_k$.
We suppose that we are able to get the value of the function and the value of the gradient but the Hessian Q is unknown.

For simplicity, we will use the following notations

- $\boldsymbol{g}_k := \nabla f(\boldsymbol{x}_k) = Q\boldsymbol{x}_k + \boldsymbol{q}$
- $\Delta \boldsymbol{g}_k := \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$
- $\Delta \boldsymbol{x}_k := \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$

It easy to see that $\Delta \boldsymbol{g}_k = Q\Delta \boldsymbol{x}_k$, *thus*

$$Q^{-1}\Delta \boldsymbol{g}_k = \Delta \boldsymbol{x}_k, \qquad \forall\{0,\ldots,k\}$$

Therefore, for the quadratic case the estimate of the inverse of the Hessian should verify the following

Property 1:

$$\boldsymbol{H}_{k+1}\Delta \boldsymbol{g}_j = \Delta \boldsymbol{x}_j, \qquad \forall j \in \{0,\ldots,k\} \tag{32}$$

Quasi-Newton methods: conditions on $\mathbf{H}_k$

Then, after n steps we have

$$\begin{aligned} \boldsymbol{H}_n \Delta \boldsymbol{g}_0 &= \Delta \boldsymbol{x}_0 \\ &\vdots \\ \boldsymbol{H}_n \Delta \boldsymbol{g}_{n-1} &= \Delta \boldsymbol{x}_{n-1} \end{aligned}$$

thus,

$$\boldsymbol{H}_n[\Delta \boldsymbol{g}_0, \ldots, \Delta \boldsymbol{g}_{n-1}] = [\Delta \boldsymbol{x}_0, \ldots, \Delta \boldsymbol{x}_{n-1}]$$

also it is easy to see that

$$Q^{-1}[\Delta \boldsymbol{g}_0, \ldots, \Delta \boldsymbol{g}_{n-1}] = [\Delta \boldsymbol{x}_0, \ldots, \Delta \boldsymbol{x}_{n-1}]$$

which shows that if property 1 *is satisfied, and* $[\Delta \boldsymbol{g}_0, \ldots, \Delta \boldsymbol{g}_{n-1}]$ *is invertible then* $\boldsymbol{H}_n = Q^{-1}$ *!*
This is quite interesting, since at iteration number $n+1$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \alpha_n \boldsymbol{H}_n \boldsymbol{g}_n \quad \Leftrightarrow \quad \boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \alpha_n \Big(D^2 f(\boldsymbol{x}_k)\Big)^{-1} \nabla f(\boldsymbol{x}_k) \tag{33}$$

Quasi-Newton Algorithm

Quasi-Newton algorithms:
with an initial condition $\boldsymbol{x}_0$ *and* $\boldsymbol{d}_0 = -\boldsymbol{H}_0 \boldsymbol{g}_0$

$$\boldsymbol{g}_k = \nabla f(\boldsymbol{x}_k) \quad (34)$$

$$\boldsymbol{d}_k = -\boldsymbol{H}_k \boldsymbol{g}_k \quad (35)$$

$$\alpha_k = \arg\min_{\alpha \geq 0} f\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k\right) \quad (36)$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k \quad (37)$$

Where $\boldsymbol{H}_0, \boldsymbol{H}_1, \ldots$ *are symmetric and satisfy Property 1 for the quadratic case*

Quasi-Newton methods: Q-conjugacy of the generated directions

From (33) we see that, for the quadratic case, at iteration $n+1$ the method is equivalent to Newton's method which converges in one step for quadratic functions.

In fact, it can be shown that for the case of quadratic functions, the algorithm converges in only n steps. This can be shown as a direct result of the following fact

Theorem 6

Consider a quasi-Newton algorithm applied to a quadratic function with Hessian $Q = Q^T$ such that property 1 is satisfied for $k \in \{0, \dots, n-1\}$, that is

$$\begin{aligned} &\boldsymbol{H}_1 \Delta \boldsymbol{g}_0 = \Delta \boldsymbol{x}_0, \\ &\boldsymbol{H}_2 \Delta \boldsymbol{g}_0 = \Delta \boldsymbol{x}_0, \ \boldsymbol{H}_2 \Delta \boldsymbol{g}_1 = \Delta \boldsymbol{x}_1, \\ &\vdots \qquad \qquad \ddots \\ &\boldsymbol{H}_n \Delta \boldsymbol{g}_0 = \Delta \boldsymbol{x}_0, \ \boldsymbol{H}_n \Delta \boldsymbol{g}_1 = \Delta \boldsymbol{x}_1, \cdots, \boldsymbol{H}_n \Delta \boldsymbol{g}_{n-1} = \Delta \boldsymbol{x}_{n-1}. \end{aligned} \tag{38}$$

where $\boldsymbol{H}_0, \boldsymbol{H}_1, \dots$ are symmetric. If $\alpha_k \neq 0$ for $i \in \{0, \dots, n-1\}$ then $\boldsymbol{d}_0, \dots, \boldsymbol{d}_{n-1}$ are Q-conjugate ($\boldsymbol{d}_k = -\boldsymbol{H}_k \boldsymbol{g}_k$).

Quasi-Newton methods: conditions on $\mathbf{H}_k$

Proof

First, remember that in this case

$$\Delta \mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i = \alpha_i \mathbf{d}_i \qquad \textit{and} \qquad \Delta \mathbf{g}_i = \mathbf{g}_{i+1} - \mathbf{g}_i = Q \Delta \mathbf{x}_i \tag{39}$$

The proof is done by induction. For $k = 1$ we have that

$$\begin{aligned}
\mathbf{d}_1^T Q \mathbf{d}_0 &= -\mathbf{g}_1^T \mathbf{H}_1 Q \mathbf{d}_0 && \textit{from } (35) \\
&= -\mathbf{g}_1^T \mathbf{H}_1 Q \frac{\Delta \mathbf{x}_0}{\alpha_0} = -\mathbf{g}_1^T \mathbf{H}_1 \frac{\Delta \mathbf{g}_0}{\alpha_0} && \textit{from } (39) \\
&= -\mathbf{g}_1^T \frac{\Delta \mathbf{x}_0}{\alpha_0} && \textit{from } (38) \textit{ and } \alpha_0 \neq 0 \\
&= -\mathbf{g}_1^T \mathbf{d}_0 && \textit{from } (39)
\end{aligned}$$

Note that

$$0 = \frac{d}{d\alpha} f(\mathbf{x}_0 + \alpha \mathbf{d}_0)\Big|_{\alpha=\alpha_0} = \left(\nabla f(\mathbf{x}_0 + \alpha_0 \mathbf{d}_0)\right)^T \mathbf{d}_0 = \left(\nabla f(\mathbf{x}_1)\right)^T \mathbf{d}_0 = \mathbf{g}_1^T \mathbf{d}_0 \tag{40}$$

Quasi-Newton methods: conditions on $\mathbf{H}_k$

Proof (Cont.)

Now we suppose that the result holds for k, that is $\boldsymbol{d}_0, \dots, \boldsymbol{d}_k$ are Q-conjugate, and to proof the case $k+1$ all we need to do is to show that $\boldsymbol{d}_{k+1}^T Q \boldsymbol{d}_i$ for $i \in \{0, \dots, k\}$

$$
\begin{aligned}
\boldsymbol{d}_{k+1}^T Q \boldsymbol{d}_i &= -\boldsymbol{g}_{k+1}^T \boldsymbol{H}_{k+1} Q \boldsymbol{d}_i \\
&= -\boldsymbol{g}_{k+1}^T \boldsymbol{H}_{k+1} Q \frac{\Delta \boldsymbol{x}_i}{\alpha_i} = -\boldsymbol{g}_{+1}^T \boldsymbol{H}_{k+1} \frac{\Delta \boldsymbol{g}_i}{\alpha_i} \qquad (\alpha_i \neq 0) \\
&= -\boldsymbol{g}_{k+1}^T \frac{\Delta \boldsymbol{x}_i}{\alpha_i} \\
&= -\boldsymbol{g}_{k+1}^T \boldsymbol{d}_i
\end{aligned}
$$

Since $\boldsymbol{d}_0, \dots, \boldsymbol{d}_k$ are Q-conjugate , then from Theorem 3 we have that $\boldsymbol{d}_{k+1}^T Q \boldsymbol{d}_i = -\boldsymbol{g}_{k+1}^T \boldsymbol{d}_i = 0$, for $i \in \{0, \dots, k\}$, which completes the proof. □

Theorem 6 shows that for the quadratic case, quasi-Newtons algorithm is a conjugate method !

As result, it solves the quadratic case in n steps (Theorem 4).

Quasi-Newton methods: conditions on $\mathbf{H}_k$

Property 2: in order to ensure that the generated directions are a decent ones, it is sufficient to impose that approximations $\mathbf{H}_k$ are symmetric positive definite

Theorem 7

Consider a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, the point $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{g}_k \neq \mathbf{0}$. Let $\mathbf{H}_k$ be a ***symmetric positive definite*** *matrix. For*

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k \mathbf{g}_k, \quad \text{with} \quad \alpha_k = \arg\min_{\alpha \geq 0} f\left(\mathbf{x}_k - \alpha \mathbf{H}_k \mathbf{g}_k\right)$$

We have that $\alpha_k > 0$, and $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.

Quasi-Newton methods: conditions on $\mathbf{H}_k$

Proof.

Consider $\boldsymbol{d} = -\boldsymbol{H}_k \boldsymbol{g}_k$, *the function* $h_k(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{d})$. *Using Taylor's theorem we have*

$$h_k(\alpha) = h_k(0) + \underbrace{\dot{h}_k(0)}_{\nabla f(\boldsymbol{x}_k)^T \boldsymbol{d} = -\boldsymbol{g}_k^T \boldsymbol{H}_k \boldsymbol{g}_k} \alpha + o(\alpha)$$

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{d}) = f(\boldsymbol{x}_k) - \alpha \boldsymbol{g}_k^T \boldsymbol{H}_k \boldsymbol{g}_k + o(\alpha)$$

Since $\boldsymbol{g}_k \neq \boldsymbol{0}$, *and* $\boldsymbol{H}_k \succ 0$, *then* $\exists \overline{\alpha} > 0$ *such that*

$$\begin{aligned} f\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}\right) &< f(\boldsymbol{x}_k), \qquad \forall \alpha \in (0, \overline{\alpha}) \\ f\left(\boldsymbol{x}_k - \alpha \boldsymbol{H}_k \boldsymbol{g}_k\right) &< f(\boldsymbol{x}_k), \qquad \forall \alpha \in (0, \overline{\alpha}) \end{aligned}$$

□

Quasi-Newton methods: determining $\mathbf{H}_k$

It is still necessary to show how to determine the matrices $\mathbf{H}_k$.

There are several algorithms that permit to determine the estimates of the inverse of the Hessian.

One example is the rank-one method which satisfy only Property 1. However, it does not guarantee the positive definiteness of the matrices $\mathbf{H}_k$.

The following algorithm uses a rank-two update method, and it is called ***Davidon–Fletcher–Powell (DFP)*** *algorithm.*

Quasi-Newton methods: the DFP Algorithm

The DFP algorithm:

with an initial condition $\mathbf{x}_0$, real symmetric positive definite matrix $\mathbf{H}_0$

$$
\begin{aligned}
\mathbf{d}_k &= -\mathbf{H}_k \mathbf{g}_k \\
\alpha_k &= \arg\min_{\alpha \geq 0} f\big(\mathbf{x}_k + \alpha \mathbf{d}_k\big) \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{d}_k \\
\Delta \mathbf{x}_k &= \mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{d}_k \\
\mathbf{g}_{k+1} &= \nabla f(\mathbf{x}_{k+1}) \\
\Delta \mathbf{g}_k &= \mathbf{g}_{k+1} - \mathbf{g}_k \\
\mathbf{H}_{k+1} &= \mathbf{H}_k + \frac{\Delta \mathbf{x}_k \Delta \mathbf{x}_k{}^T}{\Delta \mathbf{x}_k{}^T \Delta \mathbf{g}_k} - \frac{[\mathbf{H}_k \Delta \mathbf{g}_k][\mathbf{H}_k \Delta \mathbf{g}_k]^T}{\Delta \mathbf{g}_k^T \mathbf{H}_k \Delta \mathbf{g}_k}
\end{aligned}
$$

if at any iteration $\mathbf{g}_k = \mathbf{0}$ then stop.

Theorem 8

The DFP algorithm satisfies both Property 1 and Property 2.

Quasi-Newton: remarks

There are several other methods for updating $\mathbf{H}_k$ *such as the* ***BFGS*** *method developed by Broyden, Fletcher, Goldfarb and Shanno.*

Advantages of quasi-Newton methods:

- *The estimates* $\mathbf{H}_k$ *are updated iteratively*
- *Quasi-Newton methods do not rely on exact line searches for convergence. In this sense, they are more general than conjugate gradient methods*
- *Only first order derivatives are needed*
- *When* $\mathbf{H}_k$ *are definite positive, the method guarantees well defined iterations and a descent property*

Drawbacks:

- *Requires more storage and more matrix handling*