

Conjugate gradient method

Daniil Merkulov

Optimization methods. MIPT

Strongly convex quadratics

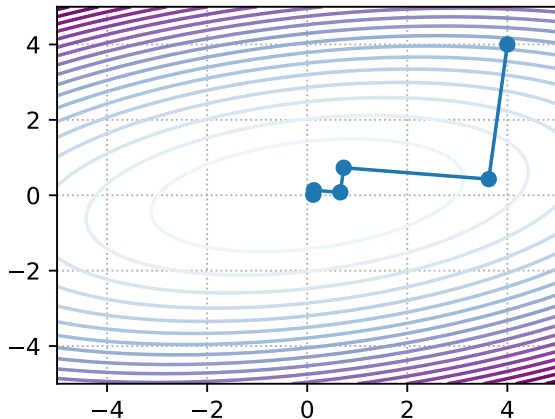
Consider the following quadratic optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) = \min_{x \in \mathbb{R}^d} \frac{1}{2} x^\top A x - b^\top x + c, \text{ where } A \in \mathbb{S}_{++}^d.$$

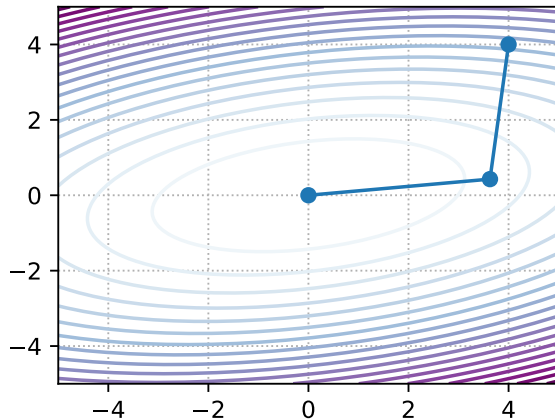
Optimality conditions

$$A x^* = b$$

Steepest Descent



Conjugate Gradient



Exact line search aka steepest descent

$$\alpha_k = \arg \min_{\alpha \in \mathbb{R}^+} f(x_{k+1}) = \arg \min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

More theoretical than practical approach. It also allows you to analyze the convergence, but often exact line search can be difficult if the function calculation takes too long or costs a lot. An interesting theoretical property of this method is that each following iteration is orthogonal to the previous one:

$$\alpha_k = \arg \min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Exact line search aka steepest descent

$$\alpha_k = \arg \min_{\alpha \in \mathbb{R}^+} f(x_{k+1}) = \arg \min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

More theoretical than practical approach. It also allows you to analyze the convergence, but often exact line search can be difficult if the function calculation takes too long or costs a lot. An interesting theoretical property of this method is that each following iteration is orthogonal to the previous one:

$$\alpha_k = \arg \min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Optimality conditions:

Exact line search aka steepest descent

$$\alpha_k = \arg \min_{\alpha \in \mathbb{R}^+} f(x_{k+1}) = \arg \min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

More theoretical than practical approach. It also allows you to analyze the convergence, but often exact line search can be difficult if the function calculation takes too long or costs a lot. An interesting theoretical property of this method is that each following iteration is orthogonal to the previous one:

$$\alpha_k = \arg \min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Optimality conditions:

$$\nabla f(x_k)^T \nabla f(x_{k+1}) = 0$$

🔥 Optimal value for quadratics

$$\nabla f(x_k)^T A(x_k - \alpha \nabla f(x_k)) - \nabla f(x_k)^T b = 0 \quad \alpha_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T A \nabla f(x_k)}$$



Figure 1: Steepest Descent

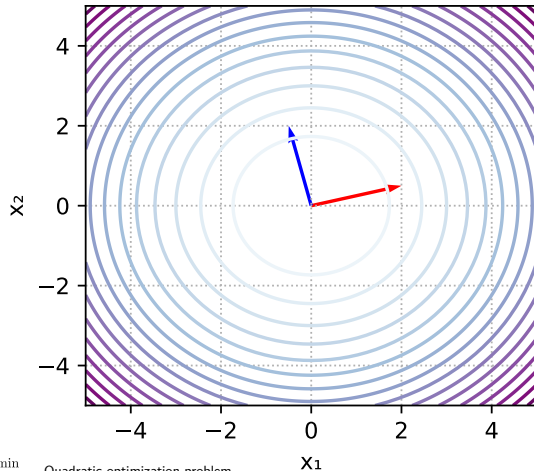
Open In Colab

Conjugate directions. A -orthogonality.

v_1 and v_2 are orthogonal

$$v_1^T v_2 = 0.00$$

$$v_1^T A v_2 = 1.19$$



\hat{v}_1 and \hat{v}_2 are A -orthogonal

$$\hat{v}_1^T \hat{v}_2 = -0.80$$

$$\hat{v}_1^T A \hat{v}_2 = -0.00$$



Conjugate directions. A -orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T I x$ looks just like on the left part of Figure 2, while in another coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A \hat{x}$, where $A \in \mathbb{S}_{++}^d$.

$$\frac{1}{2}x^T I x$$

$$\frac{1}{2}\hat{x}^T A \hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A \hat{x}$$

Conjugate directions. A -orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T Ix$ looks just like on the left part of Figure 2, while in another coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A\hat{x}$, where $A \in \mathbb{S}_{++}^d$.

$$\frac{1}{2}x^T Ix$$

$$\frac{1}{2}\hat{x}^T A\hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x}$$

Conjugate directions. A -orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T Ix$ looks just like on the left part of Figure 2, while in another coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A\hat{x}$, where $A \in \mathbb{S}_{++}^d$.

$$\frac{1}{2}x^T Ix$$

$$\frac{1}{2}\hat{x}^T A\hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x}$$

Conjugate directions. A -orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T Ix$ looks just like on the left part of Figure 2, while in another coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A\hat{x}$, where $A \in \mathbb{S}_{++}^d$.

$$\frac{1}{2}x^T Ix$$

$$\frac{1}{2}\hat{x}^T A\hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x} = \frac{1}{2}x^T Ix$$

Conjugate directions. A -orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T Ix$ looks just like on the left part of Figure 2, while in another coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A\hat{x}$, where $A \in \mathbb{S}_{++}^d$.

$$\frac{1}{2}x^T Ix$$

$$\frac{1}{2}\hat{x}^T A\hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x} = \frac{1}{2}x^T Ix \quad \text{if } x = \Lambda^{\frac{1}{2}}Q^T \hat{x}$$

Conjugate directions. A -orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T Ix$ looks just like on the left part of Figure 2, while in another coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A\hat{x}$, where $A \in \mathbb{S}_{++}^d$.

$$\frac{1}{2}x^T Ix$$

$$\frac{1}{2}\hat{x}^T A\hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x} = \frac{1}{2}x^T Ix \quad \text{if } x = \Lambda^{\frac{1}{2}}Q^T \hat{x} \text{ and } \hat{x} = Q\Lambda^{-\frac{1}{2}}x$$

Conjugate directions. A -orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T Ix$ looks just like on the left part of Figure 2, while in another coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A\hat{x}$, where $A \in \mathbb{S}_{++}^d$.

$$\frac{1}{2}x^T Ix$$

$$\frac{1}{2}\hat{x}^T A\hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x} = \frac{1}{2}x^T Ix \quad \text{if } x = \Lambda^{\frac{1}{2}}Q^T \hat{x} \text{ and } \hat{x} = Q\Lambda^{-\frac{1}{2}}x$$

Conjugate directions. A -orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T Ix$ looks just like on the left part of Figure 2, while in another coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A\hat{x}$, where $A \in \mathbb{S}_{++}^d$.

$$\frac{1}{2}x^T Ix$$

$$\frac{1}{2}\hat{x}^T A\hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x} = \frac{1}{2}x^T Ix \quad \text{if } x = \Lambda^{\frac{1}{2}}Q^T \hat{x} \text{ and } \hat{x} = Q\Lambda^{-\frac{1}{2}}x$$

A -orthogonal vectors

Vectors $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$ are called A -orthogonal (or A -conjugate) if

$$x^T Ay = 0 \quad \Leftrightarrow \quad x \perp_A y$$

When $A = I$, A -orthogonality becomes orthogonality.

Gram–Schmidt process

Idea of the method of conjugate directions

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.

Idea of the method of conjugate directions

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.
2. By the procedure of line search we find the optimal length of step. Calculate α minimizing $f(x_k + \alpha_k d_k)$ by the formula

$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top Ad_k}$$

Idea of the method of conjugate directions

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.
2. By the procedure of line search we find the optimal length of step. Calculate α minimizing $f(x_k + \alpha_k d_k)$ by the formula

$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top A d_k}$$

3. We're doing an algorithm step:

$$x_{k+1} = x_k + \alpha_k d_k$$

Idea of the method of conjugate directions

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.
2. By the procedure of line search we find the optimal length of step. Calculate α minimizing $f(x_k + \alpha_k d_k)$ by the formula

$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top Ad_k}$$

3. We're doing an algorithm step:

$$x_{k+1} = x_k + \alpha_k d_k$$

4. update the direction: $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$, where β_k is calculated by the formula:

$$\beta_k = \frac{\nabla f(x_{k+1})^\top Ad_k}{d_k^\top Ad_k}.$$

Idea of the method of conjugate directions

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.
2. By the procedure of line search we find the optimal length of step. Calculate α minimizing $f(x_k + \alpha_k d_k)$ by the formula

$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top Ad_k}$$

3. We're doing an algorithm step:

$$x_{k+1} = x_k + \alpha_k d_k$$

4. update the direction: $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$, where β_k is calculated by the formula:

$$\beta_k = \frac{\nabla f(x_{k+1})^\top Ad_k}{d_k^\top Ad_k}.$$

5. Repeat steps 2-4 until n directions are built, where n is the dimension of space (dimension of x).

Method of Conjugate Directions

If a set of vectors d_1, \dots, d_k - are A -conjugate (each pair of vectors is A -conjugate), these vectors are linearly independent. $A \in \mathbb{S}_{++}^n$.

Proof

We'll show, that if $\sum_{i=1}^k \alpha_i d_i = 0$, than all coefficients should be equal to zero:

$$\begin{aligned} 0 &= \sum_{i=1}^n \alpha_i d_i \\ &= d_j^\top A \left(\sum_{i=1}^n \alpha_i d_i \right) \\ &= \sum_{i=1}^n \alpha_i d_j^\top A d_i \\ &= \alpha_j d_j^\top A d_j + 0 + \dots + 0 \end{aligned}$$

Thus, $\alpha_j = 0$, for all other indices one have perform the same process

Conjugate Gradients

Conjugate Gradients

Conjugate Gradients

Conjugate gradient method



Conjugate Gradient = Conjugate Directions
+ Residuals as starting vectors for Gram–Schmidt

$$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$$

if \mathbf{r}_0 is sufficiently small, then return \mathbf{x}_0 as the result

$$\mathbf{d}_0 := \mathbf{r}_0$$

$$k := 0$$

repeat

$$\alpha_k := \frac{\mathbf{r}_k^\top \mathbf{r}_k}{\mathbf{d}_k^\top \mathbf{A} \mathbf{d}_k}$$

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

$$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{d}_k$$

if \mathbf{r}_{k+1} is sufficiently small, then exit loop

$$\beta_k := \frac{\mathbf{r}_{k+1}^\top \mathbf{r}_{k+1}}{\mathbf{r}_k^\top \mathbf{r}_k}$$

$$\mathbf{d}_{k+1} := \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k$$

$$k := k + 1$$

end repeat

return \mathbf{x}_{k+1} as the result

Convergence

Theorem 1. If matrix A has only r different eigenvalues, then the conjugate gradient method converges in r iterations.

Theorem 2. The following convergence bound holds

$$\|x_k - x^*\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x^*\|_A,$$

where $\|x\|_A^2 = x^\top A x$ and $\kappa(A) = \frac{\lambda_1(A)}{\lambda_n(A)}$ is the conditioning number of matrix A , $\lambda_1(A) \geq \dots \geq \lambda_n(A)$ are the eigenvalues of matrix A

Note: compare the coefficient of the geometric progression with its analog in gradient descent.

Non-linear conjugate gradient method

In case we do not have an analytic expression for a function or its gradient, we will most likely not be able to solve the one-dimensional minimization problem analytically. Therefore, step 2 of the algorithm is replaced by the usual line search procedure. But there is the following mathematical trick for the fourth point:

For two iterations, it is fair:

$$x_{k+1} - x_k = cd_k,$$

where c is some kind of constant. Then for the quadratic case, we have:

$$\nabla f(x_{k+1}) - \nabla f(x_k) = (Ax_{k+1} - b) - (Ax_k - b) = A(x_{k+1} - x_k) = cAd_k$$

Expressing from this equation the work $Ad_k = \frac{1}{c} (\nabla f(x_{k+1}) - \nabla f(x_k))$, we get rid of the “knowledge” of the function in step definition β_k , then point 4 will be rewritten as:

$$\beta_k = \frac{\nabla f(x_{k+1})^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}{d_k^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}.$$

This method is called the Polack - Ribier method.

Preconditioned conjugate gradient method