Gradient Methods for unconstrained Optimization

K.Bouanane

April 24, 2023

Introduction

Consider the problem :

$$\min_{\mathbf{x} \in U} J(\mathbf{x})$$

where $J : \mathbb{R}^n \to \mathbb{R}$

We call unconstrained optimization problem the particular case $U = \mathbb{R}^n$.

Consider the problem :

$$\min_{\mathbf{x} \in U} J(\mathbf{x})$$

where $J : \mathbb{R}^n \to \mathbb{R}$
We call unconstrained optimization problem the particular case $U = \mathbb{R}^n$.
The problem is then given by:

$$\text{Find } \mathbf{x}^* \in \mathbb{R}^n \quad \text{such that} \qquad J(\mathbf{x}^*) \leq J(\mathbf{x}), \qquad \forall\, \mathbf{x} \in \mathbb{R}^n$$

We suppose that $\mathbf{x}^*$ exists and our aim is to find an approximation of $\mathbf{x}^*$ by constructing a sequence $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ such that $\mathbf{x}^{(k)} \mapsto u^*$ for $k \mapsto +\infty$

We suppose that $\mathbf{x}^*$ exists and our aim is to find an approximation of $\mathbf{x}^*$ by constructing a sequence $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ such that $\mathbf{x}^{(k)} \mapsto u^*$ for $k \mapsto +\infty$
We aim here to give some numerical methods to construct the approximate sequence

$$\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$$

by recurrence.

We suppose that $\mathbf{x}^*$ exists and our aim is to find an approximation of $\mathbf{x}^*$ by constructing a sequence $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ such that $\mathbf{x}^{(k)} \mapsto u^*$ for $k \mapsto +\infty$

We aim here to give some numerical methods to construct the approximate sequence

$$\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$$

by recurrence.

The general expression $\mathbf{x}^{(k+1)}$ will be written as

We suppose that $\mathbf{x}^*$ exists and our aim is to find an approximation of $\mathbf{x}^*$ by constructing a sequence $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ such that $\mathbf{x}^{(k)} \mapsto u^*$ for $k \mapsto +\infty$
We aim here to give some numerical methods to construct the approximate sequence

$$\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$$

by recurrence.
The general expression $\mathbf{x}^{(k+1)}$ will be written as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho_k d^{(k)} \quad \text{such that } J(\mathbf{x}^{(k+1)}) < J(\mathbf{x}^{(k)})$$

We suppose that $\mathbf{x}^*$ exists and our aim is to find an approximation of $\mathbf{x}^*$ by constructing a sequence $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ such that $\mathbf{x}^{(k)} \mapsto u^*$ for $k \mapsto +\infty$

We aim here to give some numerical methods to construct the approximate sequence

$$\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$$

by recurrence.

The general expression $\mathbf{x}^{(k+1)}$ will be written as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho_k d^{(k)} \quad \text{such that } J(\mathbf{x}^{(k+1)}) < J(\mathbf{x}^{(k)})$$

Vectors $d^{(k)} \in \mathbb{R}^n$ are called descent directions and $\rho_k \in \mathbb{R}$ are scalars called step size (learning rate).

General scheme of unconstrained optimization algorithms.

The algorithm is of the form

Step 1: Select $\mathbf{x}^0 \in \mathbb{R}^n$ (for example $\mathbf{x}^0 = 0$). Select k_{max} the maximum number of iterations. Set k to 0.

Step 2: While(Stop condition=False) and ($k < k_{max}$) do $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \rho_k d^{(k)}$, $k \leftarrow k+1$. End while.

Step 3: If (Stop test =True) then $\mathbf{x}^{(k)}$ is an approximation of $\mathbf{x}^*$, otherwise the method doesn't converge.

As a stop condition, we often choose $\nabla J(\mathbf{x}^{(k)}) = 0$ (necessary condition for optimality).

As a stop condition, we often choose $\nabla J(\mathbf{x}^{(k)}) = 0$ (necessary condition for optimality). (In practice, $\| \nabla J(\mathbf{x}^{(k)}) \| \leq \epsilon$ where $\epsilon > 0$ is small.)

As a stop condition, we often choose $\nabla J(\mathbf{x}^{(k)}) = 0$ (necessary condition for optimality). (In practice, $\| \nabla J(\mathbf{x}^{(k)}) \| \leq \epsilon$ where $\epsilon > 0$ is small.)
Méthodes vary according to the choice of $d^{(k)}$ and ρ_k.

Gradient based optimization methods

Gradient Descent Method

Gradient Descent Method

The idea is to choose as direction of descent $d^{(k)} = -\nabla J(\mathbf{x}^{(k)})^\top$.

Gradient Descent Method

The idea is to choose as direction of descent $d^{(k)} = -\nabla J(\mathbf{x}^{(k)})^\top$. This is because $\nabla J(\mathbf{x}^{(k)})^\top$ is always orthogonal to the level curve of J in $\mathbf{x}^{(k)}$ and the function J decreases in the direction $-\nabla J(\mathbf{x}^{(k)})^\top$.

Gradient Descent Method

The idea is to choose as direction of descent $d^{(k)} = -\nabla J(\mathbf{x}^{(k)})^\top$. This is because $\nabla J(\mathbf{x}^{(k)})^\top$ is always orthogonal to the level curve of J in $\mathbf{x}^{(k)}$ and the function J decreases in the direction $-\nabla J(\mathbf{x}^{(k)})^\top$.
Using Taylor approximation, we have:

$$J(\mathbf{x}^{(k+1)}) = J(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top = J(\mathbf{x}^{(k)}) + \left\langle \nabla J(\mathbf{x}^{(k)})^\top, -\rho \nabla J(\mathbf{x}^{(k)})^\top \right\rangle + o(\rho$$

Gradient Descent Method

The idea is to choose as direction of descent $d^{(k)} = -\nabla J(\mathbf{x}^{(k)})^\top$.
This is because $\nabla J(\mathbf{x}^{(k)})^\top$ is always orthogonal to the level curve of J in $\mathbf{x}^{(k)}$ and the function J decreases in the direction $-\nabla J(\mathbf{x}^{(k)})^\top$.
Using Taylor approximation, we have:

$$J(\mathbf{x}^{(k+1)}) = J(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top = J(\mathbf{x}^{(k)}) + \left\langle \nabla J(\mathbf{x}^{(k)})^\top, -\rho \nabla J(\mathbf{x}^{(k)})^\top \right\rangle + o(\rho$$

thus

$$J(\mathbf{x}^{(k+1)}) - J(\mathbf{x}^{(k)}) = -\rho \parallel J(\mathbf{x}^{(k)}) \parallel^2 + o(\rho)$$

Gradient Descent Method

The idea is to choose as direction of descent $d^{(k)} = -\nabla J(\mathbf{x}^{(k)})^\top$. This is because $\nabla J(\mathbf{x}^{(k)})^\top$ is always orthogonal to the level curve of J in $\mathbf{x}^{(k)}$ and the function J decreases in the direction $-\nabla J(\mathbf{x}^{(k)})^\top$.

Using Taylor approximation, we have:

$$J(\mathbf{x}^{(k+1)}) = J(\mathbf{x}^{(k)} - \rho\nabla J(\mathbf{x}^{(k)})^\top = J(\mathbf{x}^{(k)}) + \left\langle \nabla J(\mathbf{x}^{(k)})^\top, -\rho\nabla J(\mathbf{x}^{(k)})^\top \right\rangle + o(\rho$$

thus

$$J(\mathbf{x}^{(k+1)}) - J(\mathbf{x}^{(k)}) = -\rho \parallel J(\mathbf{x}^{(k)}) \parallel^2 + o(\rho)$$

The right side of the equality is negative if $\rho > 0$ avec with ρ small and $\nabla J(\mathbf{x}^{(k)})^\top \neq 0$.

Gradient descent Method

The gradient descent is then described by :

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \rho_k \nabla J(\mathbf{x}^{(k)})^\top$$

where $\rho_k \in \mathbb{R}$ are selected.

Example

Compute the first two iterations for

$f(x, y) = 3x^2y - y^2x = xy(3x - y)$

$(x, y)^{(0)} = (0, 0)$

Steepest Gradient

Steepest Gradient

We aim to find ρ_k such that

$$J\left(\mathbf{x}^{(k)} - \rho_k \nabla J(\mathbf{x}^{(k)})^\top\right) = \min_{\rho \in \mathbb{R}} J\left(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top\right)$$

Case of a quadratic function

We assume that $J : \mathbb{R}^n \mapsto \mathbb{R}$ is such that

$$J(\mathbf{x}) = \frac{1}{2} \langle \mathbf{A}\,\mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{b}, \mathbf{x} \rangle + c$$

with $\mathbf{A} \in \mathbb{R}^{n \times n}$ symetric definite positive matrix , $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$.

We aim to compute ρ_k which minimizes $g : \mathbb{R} \mapsto \mathbb{R}$ given by

$$g(\rho) = J\left(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top\right)$$

ρ_k satisfies necessarily $g'(\rho_k) = 0$

We aim to compute ρ_k which minimizes $g : \mathbb{R} \mapsto \mathbb{R}$ given by

$$g(\rho) = J\left(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top\right)$$

ρ_k satisfies necessarily $g'(\rho_k) = 0$
We have $g'(\rho) = -\left\langle \nabla J\left(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top\right), \nabla J(\mathbf{x}^{(k)})^\top \right\rangle$ with $\nabla J(\mathbf{x}^{(k)})^\top = \mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b}$.

We aim to compute ρ_k which minimizes $g : \mathbb{R} \mapsto \mathbb{R}$ given by

$$g(\rho) = J\left(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top\right)$$

ρ_k satisfies necessarily $g'(\rho_k) = 0$
We have $g'(\rho) = -\left\langle \nabla J\left(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top\right), \nabla J(\mathbf{x}^{(k)})^\top \right\rangle$ with $\nabla J(\mathbf{x}^{(k)})^\top = \mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b}$.
Therefore

$$\begin{aligned} g'(\rho) &= -\left\langle \mathbf{A}\left(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top\right) - \mathbf{b}, \mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b} \right\rangle \\ &= -\left\langle (\mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b}) - \rho \mathbf{A}(\mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b}), \mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b} \right\rangle \\ &= -\parallel \mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b} \parallel^2 + \rho \left\langle \mathbf{A}(\mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b}), \mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b} \right\rangle \end{aligned}$$

Consequently

Consequently

$$g'(\rho) = 0 \Longleftrightarrow \rho = \frac{\| \mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b} \|^2}{\left\langle \mathbf{A}(\mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b}), \mathbf{A}\,\mathbf{x}^{(k)} - \mathbf{b} \right\rangle}$$

Example

Consider $f : \mathbb{R}^3 \mapsto \mathbb{R}$ such that

$$f(x_1, x_2, x_3) = \frac{3}{2}x_1^2 + 2x_2^2 + \frac{3}{2}x_3^2 + x_1x_3 + 2x_2x_3 - 3x_1 - x_3$$

et $x^0 = (0, 0, 0)^T$.

Example

Consider $f : \mathbb{R}^3 \mapsto \mathbb{R}$ such that

$$f(x_1, x_2, x_3) = \frac{3}{2}x_1^2 + 2x_2^2 + \frac{3}{2}x_3^2 + x_1x_3 + 2x_2x_3 - 3x_1 - x_3$$

et $x^0 = (0, 0, 0)^T$.

f is a quadratic function since

$$\begin{aligned}
f(x_1, x_2, x_3) &= \frac{1}{2}(3x_1^2 + 4x_2^2 + 3x_3^2) + x_1x_3 + 2x_2x_3 - 3x_1 - x_3 \\
&= \frac{1}{2}(x_1, x_2, x_3) \begin{pmatrix} 3 & 0 & 1 \\ 0 & 4 & 2 \\ 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - (3, 0, 1) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \\
&= \frac{1}{2}\langle \mathbf{A}\,\mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{b}, x \rangle
\end{aligned}$$

Example

with $\mathbf{A} = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 4 & 2 \\ 1 & 2 & 3 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$.

Conjugate Gradient Method

Notations :

1. Let $v_1, v_2, \dots, v_l \in \mathbb{R}^n$. We note $\mathcal{L}(v_1, v_2, \dots, v_l) = \{\sum_{i=1}^{l} \alpha_i v_i, \alpha_1, \alpha_2, \dots, \alpha_l \in \mathbb{R}\}$ the subspace from $\mathbb{R}^n$ spanned by $v_1, v_2, \dots, v_l$.
2. If $a \in \mathbb{R}^n$ and $M \subset \mathbb{R}^n$ then $a + M$ is the set $\{a + x, x \in M\}$

Case of a quadratic function.

Suppose that $J : \mathbb{R}^n \mapsto \mathbb{R}$ can be written

$$J(\mathbf{x}) = \frac{1}{2} \langle \mathbf{A}\,\mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{b}, \mathbf{x} \rangle + c$$

with $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a symetric positive definite matrix, $\mathbf{b} \in \mathbb{R}^n$, and $c \in \mathbb{R}$.

The main idea of the method

Recall that in Steepest descent we have

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \rho_k \nabla J(\mathbf{x}^{(k)})^\top = \min_{\rho \in \mathbb{R}} J(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top)$$

The main idea of the method

Recall that in Steepest descent we have

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \rho_k \nabla J(\mathbf{x}^{(k)})^\top = \min_{\rho \in \mathbb{R}} J(\mathbf{x}^{(k)} - \rho \nabla J(\mathbf{x}^{(k)})^\top)$$

which can be formulated as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathcal{L}(\nabla J(\mathbf{x}^{(k)})^\top)$$

is the element that minimizes J over the set $\mathbf{x}^{(k)} + \mathcal{L}(\nabla J(\mathbf{x}^{(k)})^\top)$.

The main idea of the method

For $k \in \mathbb{N}$, we note:

$$G_k = \mathcal{L}(\nabla J(\mathbf{x}^{(0)})^\top, \nabla J(\mathbf{x}^{(1)})^\top, \dots, \nabla J(\mathbf{x}^{(k)})^\top) \subset \mathbb{R}^n$$

The main idea of the method

For $k \in \mathbb{N}$, we note:

$$G_k = \mathcal{L}(\nabla J(\mathbf{x}^{(0)})^\top, \nabla J(\mathbf{x}^{(1)})^\top, \dots, \nabla J(\mathbf{x}^{(k)})^\top) \subset \mathbb{R}^n$$

Conjugate gradients aims to find $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + G_k$ such that

$$J(\mathbf{x}^{k+1}) = \min_{v \in \mathbf{x}^{(k)} + G_k} J(v)$$

The main idea of the method

For $k \in \mathbb{N}$, we note:

$$G_k = \mathcal{L}(\nabla J(\mathbf{x}^{(0)})^\top, \nabla J(\mathbf{x}^{(1)})^\top, \ldots, \nabla J(\mathbf{x}^{(k)})^\top) \subset \mathbb{R}^n$$

Conjugate gradients aims to find $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + G_k$ such that

$$J(\mathbf{x}^{k+1}) = \min_{v \in \mathbf{x}^{(k)} + G_k} J(v)$$

We therefore minimize in a larger space than in the optimal step gradient method. We then expect to find the minimum more quickly.

Conjugate Gradient Algorithm for quadratic function.

Step 1: We set $k = 0$ and choose $\mathbf{x}^{(0)} \in \mathbb{R}^n$.
$d^{(0)} = \nabla J(\mathbf{x}^{(0)})^\top := \mathbf{A}\,\mathbf{x}^{(0)} - \mathbf{b}$.

Step 2: If $\nabla J(\mathbf{x}^{(k)})^\top = 0$ then Stop. The optimal solution is $\mathbf{x}^{(k)}$. Else go to Step 3.

Step 3: We set

$$
\begin{aligned}
\rho_k &= -\frac{\left\langle \nabla J(\mathbf{x}^{(k)})^\top, d^{(k)} \right\rangle}{\left\langle \mathbf{A} d^{(k)}, d^{(k)} \right\rangle} \\
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \rho_k d^{(k)} \\
\beta_k &= \frac{\| \nabla J(\mathbf{x}^{(k+1)})^\top \|^2}{\| \nabla J(\mathbf{x}^{(k)})^\top \|^2} \\
d^{(k+1)} &= \nabla J(\mathbf{x}^{(k+1)})^\top + \beta_k d^{(k)}
\end{aligned}
$$

$k \leftarrow k + 1$, Go to Step 2.

Example

Let $J : \mathbb{R}^2 \mapsto \mathbb{R}$ be such that

$$J(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$$

and $x^{(0)} = (1, 1)$.