

Gradient descent : with optimal step

Definition

Let f be a convex and continuously differentiable function on \mathbb{R}^n . We say that f is **strongly convex or α -elliptical** if it exists $\alpha > 0$ such that

$$\langle \nabla f(v) - \nabla f(u), v - u \rangle \geq \alpha \|v - u\|, \quad \forall u, v \in \mathbb{R}^n$$

What can we say about $\langle \nabla f(u_{k+1}), \nabla f(u_k) \rangle$ based on
 $\rho_k = \arg \min_{\rho > 0} f(u_k - \rho d_k)$?

Gradient descent : with optimal step

If ρ_k minimize $f(u_k - \rho_k d_k)$ we have :

$$\frac{\partial}{\partial \rho} f(u_k - \rho \nabla f(u_k)) \Big|_{\rho=\rho_k} = 0,$$

$$\iff \langle \nabla f(u_k - \rho_k \nabla f(u_k)), \nabla f(u_k) \rangle = 0,$$

$$\iff \langle \nabla f(u_{k+1}), \nabla f(u_k) \rangle = 0.$$

The last equality is called the **optimality condition**.

Proposition

If f is a **strongly convex** then GD with optimal step converges

Gradient descent : with optimal step

$$\begin{aligned}
 \Rightarrow \langle Au_k - b, Au_k - b - \rho_k A(Au_k - b) \rangle &= 0 \\
 \Rightarrow \langle Au_k - b, Au_k - b \rangle &= \langle Au_k - b, \rho_k A(Au_k - b) \rangle \\
 \Rightarrow \rho_k &= \frac{\langle Au_k - b, Au_k - b \rangle}{\langle Au_k - b, A(Au_k - b) \rangle}
 \end{aligned}$$

We finally have the following algorithm :

- Initialize $u_0 \in \mathbb{R}^n$
- At each step, calculate $\rho_k = \frac{\|Au_k - b\|^2}{\|Au_k - b\|_A^2}$
- Set $u_{k+1} = u_k - \rho_k(Au_k - b)$
- Stop if $\|\nabla J(u_{k+1})\| = \|Au_{k+1} - b\| \leq \epsilon$

Gradient descent : with optimal step

Exercise

Consider the matrices $A = \begin{pmatrix} 6 & 2 \\ 2 & 4 \end{pmatrix}$ and $b = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ and the application f defined by $f(v) = \langle Av, v \rangle + \langle b, v \rangle$

1. Explain why f is convex.
2. Solve the problem $u = \arg \min_{v \in \mathbb{R}^2} f(v)$.
3. For a given vector u_k , calculate ∇f_{u_k} and ρ_k .
4. Implement the presented method to solve this problem.

Correction

- f is defined as a quadratic function where A is PSD, so f is convex.
- We have to solve :

$$J_{f(x,y)} = (12x + 4y + 2, \quad 4x + 8y + 3) = (0, 0).$$

The solution is $\left(-\frac{1}{20}, -\frac{7}{20} \right)$.

- Let set $u_k = (v_1, v_2)$ then :

$$\nabla f_{u_k} = (12v_1 + 4v_2 + 2, \quad 4v_1 + 8v_2 + 3),$$

and $\rho_k = \frac{\|2Au_k + b\|_2^2}{\|2Au_k + b\|_A^2}$

Exercise

Let f be the function defined by : $f(x, y) = 4x^2 - 4xy + 2y^2$.

1. Is the function f convex ?
2. Apply the gradient descent with optimal step to calculate the three first steps of the algorithm using $(x_0, y_0) = (1, 1)$.

Correction 1/3

- The function f can be rewritten as : $f(u) = \frac{1}{2}u^T Au - b^T u$, where $b = (0, 0)^T$ and $A = \begin{pmatrix} 8 & -4 \\ -4 & 4 \end{pmatrix}$. The function f is a quadratic function, furthermore the matrix A is PSD so the function f is convex.
- The optimal learning rate is given by :

$$\rho_k \frac{\|Au_k - b\|_2^2}{\|Au_k - b\|_A^2},$$

where the matrix A and the vector b were previously introduced.

Correction 2/3

- The function f can be rewritten as : $f(u) = \frac{1}{2}u^T Au - b^T u$, where $b = (0 \ 0)^T$ and $A = \begin{pmatrix} 8 & -4 \\ -4 & 4 \end{pmatrix}$. The function f is a quadratic function, furthermore the matrix A is PSD so the function f is convex.
- The optimal learning rate is given by :

$$\rho_k = \frac{\|Au_k - b\|_2^2}{\|Au_k - b\|_A^2},$$

where the matrix A and the vector b were previously introduced.
Recall that the process is defined by :

$$u_{k+1} = u_k - \rho_k \nabla f(u_k).$$

We will now apply this process to compute the three first iterations.

Correction 3/3

1. For the first iteration : $\rho_0 = \frac{\|Au_0\|_2^2}{\|Au_0\|_A^2} = \frac{16}{128} = \frac{1}{8}$. And
 $\nabla f(u_0) = Au_0 = (4 \ 0)^T$.

$$u_1 = (1 \ 1)^T - \frac{1}{8}(4 \ 0)^T = (0.5 \ 1)^T.$$

2. For the second iteration : $\nabla f(u_1) = Au_1 = (0 \ 2)^T$. The learning rate is given by : $\rho_1 = \frac{\|Au_1\|_2^2}{\|Au_1\|_A^2} = \frac{4}{16} = \frac{1}{4}$. Thus u_2 is given by :

$$u_2 = (0.5 \ 1)^T - \frac{1}{4}(0 \ 2)^T = (0.5 \ 0.5)^T.$$

3. For the third iteration : $\nabla f(u_2) = Au_2 = (2 \ 0)^T$. The learning rate is given by : $\rho_2 = \frac{\|Au_2\|_2^2}{\|Au_2\|_A^2} = \frac{4}{32} = \frac{1}{8}$. Thus u_3 is given by :

$$u_3 = (0.5 \ 0.5)^T - \frac{1}{8}(2 \ 0)^T = (0.25 \ 0.5)^T.$$

Gradient Descent : Armijo Criterium

Idea : use a linear search to find the **learning rate**.

Given a $\theta \in]0, 1[$, choose the greatest ρ such that :

$$f(u_k - \rho \nabla f(u_k)) \leq f(u_k) - \theta \rho \|\nabla f(u_k)\|^2.$$

At each step, we reduce the function's value of at least $\theta \|\nabla f(u_k)\|^2$.

Gradient Descent : Armijo Criterium

Idea : use a linear search to find the **learning rate**.

Given a $\theta \in]0, 1[$, choose the greatest ρ such that :

$$f(u_k - \rho \nabla f(u_k)) \leq f(u_k) - \theta \rho \|\nabla f(u_k)\|^2.$$

At each step, we reduce the function's value of at least $\theta \|\nabla f(u_k)\|^2$.

Armijo's condition :

- ▶ Choose $\alpha_0 > 0$ and $0 < \theta < 1$,
- ▶ Choose the greatest $s \in \mathbb{Z}$ such that :

$$f(u_k - \alpha_0 2^s \nabla f(u_k)) \leq f(u_k) - 2^s \alpha_0 \theta \|\nabla f(u_k)\|^2.$$

- ▶ Set $u_{k+1} \leftarrow u_k - \alpha_0 2^s \nabla f(u_k)$.

Gradient Descent : Armijo Criterium and Wolfe's Criteria

Theorem

If the function f is **strictly convex** and if its gradient ∇f is **Lipschitz**, then the Armijo's algorithm **converge**.

If we add the following condition to the previous one, given $0 < \theta < \eta < 1$:

$$\langle \nabla f(u_k), \nabla f(u_k - \rho \nabla f(u_k)) \rangle \geq \eta \|\nabla f(u_k)\|^2,$$

we get the **Wolfe's Criteria**

Conjugate Gradient

Definition

Let A be a **symmetric PD** matrix and u, v two vectors. u, v are **conjugate** with respect to A if

$$\langle Au, v \rangle = 0$$

Conjugate Gradient

Definition

Let A be a **symmetric PD** matrix and u, v two vectors. u, v are **conjugate** with respect to A if

$$\langle Au, v \rangle = 0$$

Let A be a **symmetric PD** matrix and f the function defined by

$$f(v) = \frac{1}{2} \langle Av, v \rangle - \langle b, v \rangle.$$

The objective is to build a series of **conjugate descent direction**

Conjugate Gradient

- Let $u_0 \in \mathbb{R}^n$, define a first direction of descent $d_0 = \nabla f(u_0)$ and minimize f along this direction :

$$\arg \min_{\alpha_0} f(u_0 - \alpha_0 d_0).$$

- Solving this problem we get :

$$\alpha_0 = \frac{\langle \nabla f(u_0), d_0 \rangle}{\langle A d_0, d_0 \rangle}.$$

- We set $u_1 = u_0 - \alpha_0 d_0$
- To build $d_1 = \nabla f(u_1) + \beta_0 d_0$, we need to find the value of $\beta_0 \in \mathbb{R}$ such that

$$\langle A d_1, d_0 \rangle = 0.$$

Conjugate Gradient

- We then have to solve $\langle A\nabla f(u_1), d_0 \rangle + \langle A\beta_0 d_0, d_0 \rangle = 0$. The solution is given by

$$\beta_0 = -\frac{\langle A\nabla f(u_1), d_0 \rangle}{\langle Ad_0, d_0 \rangle}.$$

Once it's done, you'll do as before.

You set $\alpha_1 = \arg \min_{\alpha} f(u_1 - \alpha d_1)$.

Set $u_2 = u_1 - \alpha_1 d_1$. And so on ...

Conjugate Gradient : Summary

Algorithm :

- ▶ Choose $u_0 \in \mathbb{R}^n$ and $d_0 = \nabla f(u_0)$.
- ▶ Set $\alpha_0 = \frac{\langle \nabla f(u_0), d_0 \rangle}{\langle Ad_0, d_0 \rangle}$ and $u_1 = u_0 - \alpha_0 d_0$.
- ▶ $\beta_0 = -\frac{\langle A\nabla f(u_1), d_0 \rangle}{\langle Ad_0, d_0 \rangle}$.

For $k \geq 1$ do,

- ▶ Set $d_k = \nabla f(u_k) + \beta_{k-1} d_{k-1}$.
- ▶ Set $\alpha_k = \frac{\langle \nabla f(u_k), d_k \rangle}{\langle Ad_k, d_k \rangle}$ and $u_{k+1} = u_k - \alpha_k d_k$.
- ▶ Set $\beta_k = \frac{\langle A\nabla f(u_{k+1}), d_k \rangle}{\langle Ad_k, d_k \rangle}$

Untill $\|\nabla f(u_{k+1})\| \leq \varepsilon$.

Conjugate Gradient : Results

Proposition

For all $1 \leq k \leq n$ such that $\nabla f(u_0), \dots, \nabla f(u_n)$ are non equal to zero, we have the following relations for all $0 \leq l \leq k - 1$:

$$\langle \nabla f(u_k), \nabla f(u_l) \rangle = 0$$

and

$$\langle Ad_k, d_l \rangle = 0.$$

Theorem

If A is a symmetric positive and definite matrix, then the conjugate gradient method converges with **at most n steps**.

Try to prove the proposition by induction at home

Newton's Method

The **Newton's Method** is a gradient descent algorithm that refines the direction of the descent as follows :

$$u_{k+1} \leftarrow u_k - (\nabla^2 f(u_k))^{-1} \cdot \nabla f(u_k).$$

- ✓ Requires less iterations to converge
- ✗ Requires the inverse of the Hessian of the function we want to optimize ($\Theta(n^3)$).
- ✗ The Hessian is not always invertible at a given point.

Newton's Method

For Logistic Regression, we want to maximize $l(x, a)$ with a **possible** solution given by solving the equation :

$$\nabla_a l(x, a) = \nabla_a \left(\sum_{i=1}^n y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right) = 0,$$

where $p = (1 + \exp(-a^T x))^{-1}$.

Explain why the log-likelihood is **concave**. Calculate the **first and second derivatives** of the function l .

