Introduction to Optimization

Note 7

Jun Moon

junmoon@hanyang.ac.kr

Outline

- Gradient Descent

Summary of the gradient descent method

Unconstrained Minimization

$$\min_{x \in \mathbb{R}^n} f(x)$$

- Assumption
  - f is convex and continuously differentiable
  - The optimal value $f^* = \inf_{x \in \mathbb{R}^n} f(x)$ is finite
- Minimization methods
  - Iterative methods for the form

    $$x_{k+1} = x_k + \alpha_k d_k, \ x_0 = x \in \mathbb{R}^n$$

    α_k: step size
    d_k: direction of the iterative algorithm
  - Generate a sequence of points $\{x_k\}$ such that $f(x_k) \to f^*$ as $k \to \infty$
  - Can be interpreted as iterative methods for solving the system of equations to satisfy the necessary and sufficient optimality condition

    $$\nabla f(x^*) = 0$$

Unconstrained Minimization: Gradient Descent Method

- Gradient descent algorithm

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \; x_0 = x$$

- α_k: stepsize
  - Constant
  - Diminishing: $\alpha_k \to 0$ with $\sum_{k=1}^{\infty} \alpha_k = \infty$
  - Linear search types (optimal stepsize, hard to find)
    - Exact line search: $\alpha_k = \arg\min_{\alpha > 0} f(x_k + \alpha d_k)$
    - Backtracking line search

Gradient Descent with Bounded Gradients

Theorem

Suppose that the gradient is bounded, that is, for some $L > 0$

$$\|\nabla f(x)\| \leq L \quad \forall x, y \in \mathbb{R}^n.$$

Let the stepsize be constant, i.e., $\alpha_k = \alpha$. Then the gradient descent algorithm generates the sequence $\{x_k\}$ such that

$$\liminf_{k \to \infty} f(x_k) \leq f^* + \frac{\alpha L}{2}$$

Also, when diminishing step size is used, i.e., $\sum_{k=0}^{\infty} \alpha_k = \infty$, $\lim_{k \to \infty} f(x_k) = f^*$.

Proof: Assume that the result does not hold, i.e., for some $\hat{y}$ with $f(\hat{y}) = f^* + \epsilon$,

$$f(x_k) - f(\hat{y}) \geq \frac{\alpha L}{2} + \epsilon, \ \forall k.$$

Then we can show that $\|x_k - \hat{y}\|^2 \leq \|x_0 - \hat{y}\|^2 - 2k\alpha\epsilon$, which fails to hold when k is sufficiently large.

Gradient Descent with the Lipschitz Gradient

Theorem

Suppose that the gradient of f is Lipschitz continuous, i.e., for some $M > 0$

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\| \quad \forall x, y \in \mathbb{R}^n$$

Then for the constant stepsize $\alpha \leq \frac{2}{M}$, we have

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$$

Furthermore, if X^* is nonempty, then the gradient descent algorithm converges to the optimal point.

Proof: By using the Lipschitz constant,

$$f(x_{k+1}) \leq f(x_k) - \frac{\alpha}{2}(2 - \alpha M)\|f(x_k)\|^2.$$

Then with $2 - M\alpha \geq 0$, $\sum_{k=1}^{\infty} \|\nabla^2 f(x_k)\|^2 < \infty$.

Gradient Descent with Strong Convexity

Theorem

Suppose that f is strongly convex, i.e., $mI \leq \nabla^2 f(x) \leq MI$ for all x. Then with the constant stepsize $\alpha < \frac{\min(2,m)}{M}$,

$$\|x_k - x^*\| \leq cq^k \quad 0 < q < 1$$

That is, the gradient descent algorithm has the geometric convergence rate.

Using the strong convexity assumption,

$$\|x_{k+1} - x^*\|^2 \leq (1 - m\alpha + \alpha^2 M)^{k+1} \|x_0 - x^*\|^2$$

- Geometric convergence is not bad, but there are not many functions that satisfy the strong convexity assumption
- Note that with the strongly convexity, x^* is unique

Steepest Gradient Descent

Fasted Gradient Method

- optimizing the step size

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)}).$$

$$\alpha_k = \arg\min_{\alpha \geq 0} f(x^{(k)} - \alpha \nabla f(x^{(k)})).$$

Proposition 8.1 *If $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ is a steepest descent sequence for a given function $f : \mathbb{R}^n \to \mathbb{R}$, then for each k the vector $\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ is orthogonal to the vector $\boldsymbol{x}^{(k+2)} - \boldsymbol{x}^{(k+1)}$.* □

Proposition 8.2 *If $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ is the steepest descent sequence for $f : \mathbb{R}^n \to \mathbb{R}$ and if $\nabla f(\boldsymbol{x}^{(k)}) \neq \mathbf{0}$, then $f(\boldsymbol{x}^{(k+1)}) < f(\boldsymbol{x}^{(k)})$.* □

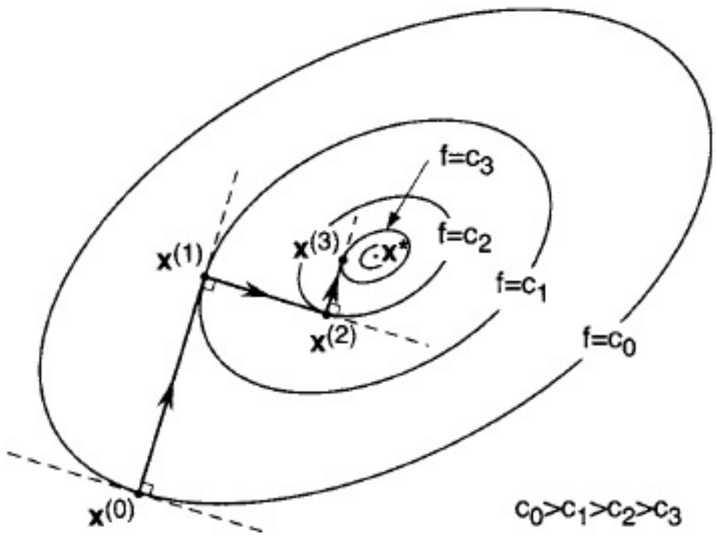


Figure 8.2 Typical sequence resulting from the method of steepest descent.

Several different stopping criteria

$$|f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^{(k)})| < \varepsilon,$$

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| < \varepsilon.$$

$$\frac{|f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^{(k)})|}{|f(\boldsymbol{x}^{(k)})|} < \varepsilon$$

$$\frac{\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|}{\|\boldsymbol{x}^{(k)}\|} < \varepsilon.$$

Example 8.1 We use the method of steepest descent to find the minimizer of

$$f(x_1, x_2, x_3) = (x_1 - 4)^4 + (x_2 - 3)^2 + 4(x_3 + 5)^4.$$

The initial point is $\boldsymbol{x}^{(0)} = [4, 2, -1]^\top$. We perform three iterations.

We find that

$$\nabla f(\boldsymbol{x}) = [4(x_1 - 4)^3, 2(x_2 - 3), 16(x_3 + 5)^3]^\top.$$

Hence,

$$\nabla f(\boldsymbol{x}^{(0)}) = [0, -2, 1024]^\top.$$

To compute $\boldsymbol{x}^{(1)}$, we need

$$\begin{aligned}\alpha_0 &= \arg\min_{\alpha \geq 0} f(\boldsymbol{x}^{(0)} - \alpha \nabla f(\boldsymbol{x}^{(0)})) \\ &= \arg\min_{\alpha \geq 0} (0 + (2 + 2\alpha - 3)^2 + 4(-1 - 1024\alpha + 5)^4) \\ &= \arg\min_{\alpha \geq 0} \phi_0(\alpha).\end{aligned}$$

$$\alpha_0 = 3.967 \times 10^{-3}.$$

For illustrative purpose, we show a plot of $\phi_0(\alpha)$ versus α in Figure 8.3, obtained using MATLAB. Thus,

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \alpha_0 \nabla f(\boldsymbol{x}^{(0)}) = [4.000, 2.008, -5.062]^\top.$$

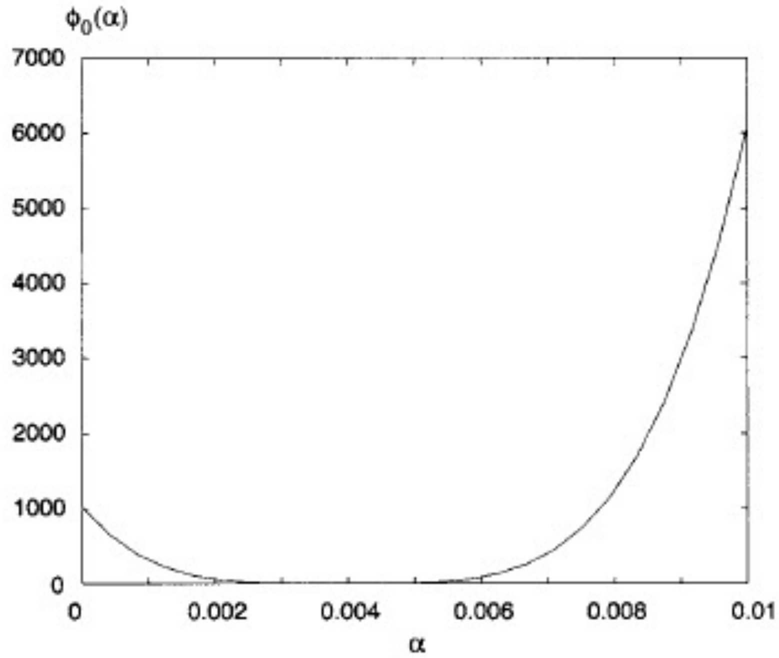


Figure 8.3 Plot of $\phi_0(\alpha)$ versus α.

Special case: Quadratic Optimization (convex optimization)

- Sometime called Quadratic Programming (QP)

$$f(x) = \frac{1}{2} x^\top Q x - b^\top x$$

$$\nabla f(x) = Qx - b$$

$$Q = Q^\top > 0, \; b \in \mathbb{R}^n$$

Convex!!!

$$f(x) = \frac{1}{2}x^\top Q x - b^\top x$$

$$\nabla f(x) = Qx - b$$

$$Q = Q^\top > 0,\ b \in \mathbb{R}^n$$

$$\nabla f(x) = Qx - b = 0 \text{ 1st-order condition}$$

$$f(x^*) = 0 \ \Leftrightarrow \ x^* = Q^{-1}b$$

Optimal solution

The Hessian of f is $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{Q} = \boldsymbol{Q}^\top > 0$. To simplify the notation we write $\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)})$. Then, the steepest descent algorithm for the quadratic function can be represented as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)},$$

where

$$\begin{aligned}
\alpha_k &= \underset{\alpha \geq 0}{\arg\min}\, f(\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)}) \\
&= \underset{\alpha \geq 0}{\arg\min} \left(\frac{1}{2} (\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)})^\top \boldsymbol{Q} (\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)}) - (\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)})^\top \boldsymbol{b} \right).
\end{aligned}$$

algorithm stops. Because $\alpha_k \geq 0$ is a minimizer of $\phi_k(\alpha) = f(\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)})$, we apply the FONC to $\phi_k(\alpha)$ to obtain

$$\phi_k'(\alpha) = (\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)})^\top \boldsymbol{Q}(-\boldsymbol{g}^{(k)}) - \boldsymbol{b}^\top(-\boldsymbol{g}^{(k)}).$$

Therefore, $\phi_k'(\alpha) = 0$ if $\alpha \boldsymbol{g}^{(k)\top} \boldsymbol{Q} \boldsymbol{g}^{(k)} = (\boldsymbol{x}^{(k)\top} \boldsymbol{Q} - \boldsymbol{b}^\top) \boldsymbol{g}^{(k)}$. But

$$\boldsymbol{x}^{(k)\top} \boldsymbol{Q} - \boldsymbol{b}^\top = \boldsymbol{g}^{(k)\top}.$$

Hence,

$$\alpha_k = \frac{\boldsymbol{g}^{(k)\top} \boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)\top} \boldsymbol{Q} \boldsymbol{g}^{(k)}}.$$

Steepest Gradient Descent for QP

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)}).$$

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\boldsymbol{x}^{(k)} - \alpha \nabla f(\boldsymbol{x}^{(k)})).$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \frac{\boldsymbol{g}^{(k)\top} \boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)\top} \boldsymbol{Q} \boldsymbol{g}^{(k)}} \boldsymbol{g}^{(k)},$$

$$\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)}) = \boldsymbol{Q}\boldsymbol{x}^{(k)} - \boldsymbol{b}.$$

Analysis

- Does the steepest gradient descent for QP converge?

- If it converges, what is the convergence rate?

Not necessarily steepest gradient descent

- General gradient descent for QP

Theorem 8.1 *Let $\{\boldsymbol{x}^{(k)}\}$ be the sequence resulting from a gradient algorithm $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{g}^{(k)}$. Let γ_k be as defined in Lemma 8.1, and suppose that $\gamma_k > 0$ for all k. Then, $\{\boldsymbol{x}^{(k)}\}$ converges to $\boldsymbol{x}^*$ for any initial condition $\boldsymbol{x}^{(0)}$ if and only if*

$$\sum_{k=0}^{\infty} \gamma_k = \infty.$$

□

$$\gamma_k = \alpha_k \frac{\boldsymbol{g}^{(k)\top} \boldsymbol{Q} \boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)\top} \boldsymbol{Q}^{-1} \boldsymbol{g}^{(k)}} \left(2 \frac{\boldsymbol{g}^{(k)\top} \boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)\top} \boldsymbol{Q} \boldsymbol{g}^{(k)}} - \alpha_k \right).$$

$$\lambda_{\min}(\boldsymbol{Q})\|\boldsymbol{x}\|^2 \leq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} \leq \lambda_{\max}(\boldsymbol{Q})\|\boldsymbol{x}\|^2,$$

where $\lambda_{\min}(\boldsymbol{Q})$ denotes the minimal eigenvalue of $\boldsymbol{Q}$ and $\lambda_{\max}(\boldsymbol{Q})$ denotes the maximal eigenvalue of $\boldsymbol{Q}$. For $\boldsymbol{Q} = \boldsymbol{Q}^\top > 0$, we also have

$$\lambda_{\min}(\boldsymbol{Q}^{-1}) = \frac{1}{\lambda_{\max}(\boldsymbol{Q})},$$
$$\lambda_{\max}(\boldsymbol{Q}^{-1}) = \frac{1}{\lambda_{\min}(\boldsymbol{Q})},$$

$$\lambda_{\min}(\boldsymbol{Q}^{-1})\|\boldsymbol{x}\|^2 \leq \boldsymbol{x}^\top \boldsymbol{Q}^{-1}\boldsymbol{x} \leq \lambda_{\max}(\boldsymbol{Q}^{-1})\|\boldsymbol{x}\|^2.$$

Convergence!!!!!!

Theorem 8.2 *In the steepest descent algorithm, we have* $\boldsymbol{x}^{(k)} \to \boldsymbol{x}^*$ *for any* $\boldsymbol{x}^{(0)}$. □

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \frac{\boldsymbol{g}^{(k)\top}\boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)\top}\boldsymbol{Q}\boldsymbol{g}^{(k)}}\boldsymbol{g}^{(k)},$$

$$\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)}) = \boldsymbol{Q}\boldsymbol{x}^{(k)} - \boldsymbol{b}.$$

Not necessarily steepest gradient descent

- General gradient descent for QP

Theorem 8.3 *For the fixed-step-size gradient algorithm, $\boldsymbol{x}^{(k)} \to \boldsymbol{x}^*$ for any $\boldsymbol{x}^{(0)}$ if and only if*

$$0 < \alpha < \frac{2}{\lambda_{\max}(\boldsymbol{Q})}.$$

□

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)}).$$

$$\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)}) = \boldsymbol{Q}\boldsymbol{x}^{(k)} - \boldsymbol{b}.$$

Example 8.4 Let the function f be given by

$$f(\boldsymbol{x}) = \boldsymbol{x}^\top \begin{bmatrix} 4 & 2\sqrt{2} \\ 0 & 5 \end{bmatrix} \boldsymbol{x} + \boldsymbol{x}^\top \begin{bmatrix} 3 \\ 6 \end{bmatrix} + 24.$$

We wish to find the minimizer of f using a fixed-step-size gradient algorithm

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha \nabla f(\boldsymbol{x}^{(k)}),$$

where $\alpha \in \mathbb{R}$ is a fixed step size.

To apply Theorem 8.3, we first symmetrize the matrix in the quadratic term of f to get

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \begin{bmatrix} 8 & 2\sqrt{2} \\ 2\sqrt{2} & 10 \end{bmatrix} \boldsymbol{x} + \boldsymbol{x}^\top \begin{bmatrix} 3 \\ 6 \end{bmatrix} + 24.$$

The eigenvalues of the matrix in the quadratic term are 6 and 12. Hence, using Theorem 8.3, the algorithm converges to the minimizer for all $\boldsymbol{x}^{(0)}$ if and only if α lies in the range $0 < \alpha < 2/12$. ∎

Theorem 8.6 *Let $\{\boldsymbol{x}^{(k)}\}$ be a convergent sequence of iterates of the steepest descent algorithm applied to a function f. Then, the order of convergence of $\{\boldsymbol{x}^{(k)}\}$ is 1 in the worst case; that is, there exist a function f and an initial condition $\boldsymbol{x}^{(0)}$ such that the order of convergence of $\{\boldsymbol{x}^{(k)}\}$ is equal to 1.* $\square$

$$\mathcal{O}\left(\frac{1}{k}\right) \qquad \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\| = O(\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|^p), \quad \text{p=1}$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \frac{\boldsymbol{g}^{(k)\top}\boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)\top}\boldsymbol{Q}\boldsymbol{g}^{(k)}}\boldsymbol{g}^{(k)},$$

$$\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)}) = \boldsymbol{Q}\boldsymbol{x}^{(k)} - \boldsymbol{b}.$$

Nesterov Accelerated Gradient Descent

Yurii Nesterov

From Wikipedia, the free encyclopedia

Yurii Nesterov is a Russian mathematician, an internationally recognized expert in convex optimization, especially in the development of efficient algorithms and numerical optimization analysis. He is currently a professor at the University of Louvain (UCLouvain).

Contents [hide]

Biography [edit]

In 1977, Yurii Nesterov graduated in applied mathematics at Moscow State University. From 1977 to 1992 he was a researcher at the Central Economic Mathematical Institute of the Russian Academy of Sciences. Since 1993, he has been working at UCLouvain, specifically in the Department of Mathematical Engineering from the Louvain School of Engineering, Center for Operations Research and Econometrics.

In 2000, Nesterov received the Dantzig Prize.[1]

In 2009, Nesterov won the John von Neumann Theory Prize.[2]

In 2016, Nesterov received the EURO Gold Medal.[3]

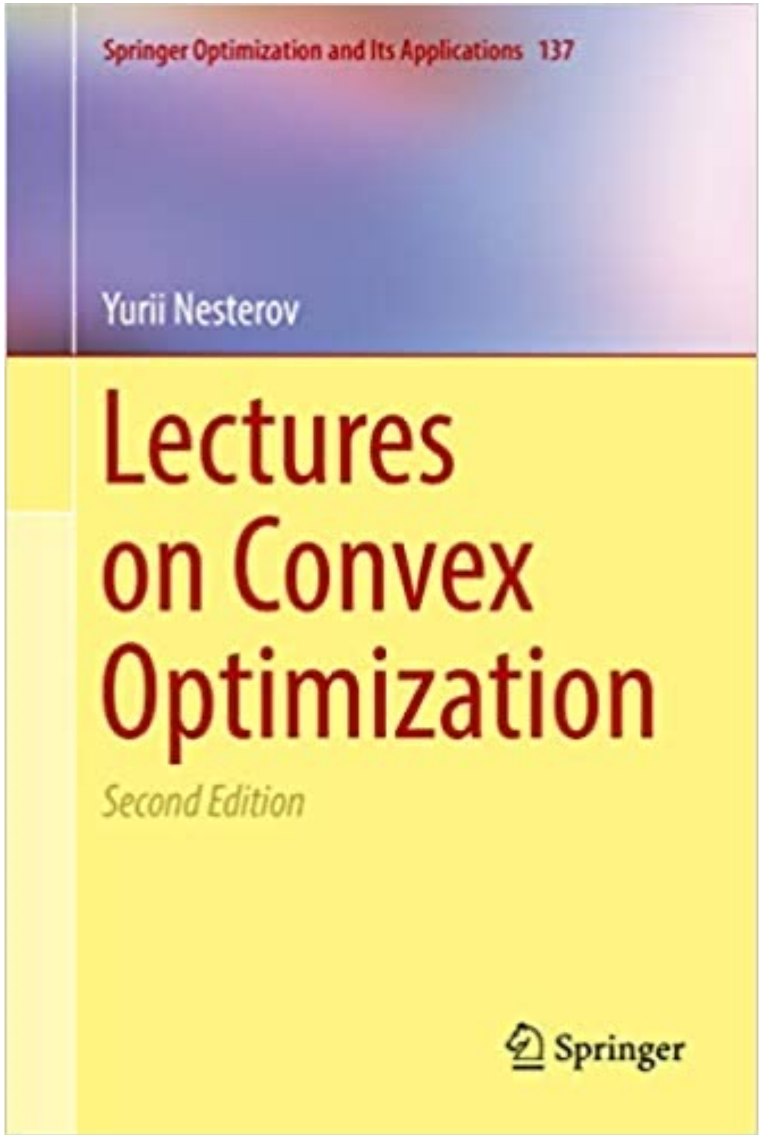
Yurii Nesterov

2005 in Oberwolfach

Born	January 25, 1956 (age 65) Moscow, USSR
Citizenship	Belgium
Alma mater	Moscow State University (1977)
Awards	Dantzig Prize, 2000 John von Neumann Theory Prize, 2009 EURO Gold Medal, 2016

Scientific career

Fields	Convex optimization,


Springer Optimization and Its Applications 137
Yurii Nesterov
Lectures on Convex Optimization
Second Edition
Springer

(Recall) Special case: Quadratic Optimization (convex optimization)

- Sometime called Quadratic Programming (QP)

$$f(x) = \frac{1}{2}x^\top Qx - b^\top x$$

$$\nabla f(x) = Qx - b$$

$$Q = Q^\top > 0, \; b \in \mathbb{R}^n$$

Convex!!!

Nesterov's acceleration

$$\begin{array}{rl} \text{GD} & \mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k) \\ \text{HBM} & \mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k) + \beta_k(\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \text{Nesterov} & \mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k + \beta_k(\mathbf{x}_k - \mathbf{x}_{k-1})) + \beta_k(\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \text{Nesterov-2} & \mathbf{y}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k) \\ & \mathbf{x}_{k+1} = \mathbf{y}_{k+1} + \beta_k(\mathbf{y}_{k+1} - \mathbf{y}_k) \end{array}$$

We saw HBM with fixed β.

Nesterov gave the update scheme with *close-form formula* for β_k (in 1983)

$$\alpha_1 \in [0,1], \ \alpha_{k+1} = \frac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}, \ \beta_k = \frac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}.$$

Note: β_k is not fix, it is a function of α_1. We need to guess α_1.

How to get Nesterov-2 from Nesterov : set $\mathbf{x}_{-1} = \mathbf{x}_0$, $\mathbf{y}_{-1} = \mathbf{y}_0$

Algorithm: Nesterov's accelerated gradient for $(\mathcal{P})$

Result: A solution $\mathbf{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\mathbf{x}_0 \in \mathbb{R}^n$ $\alpha_1 \in (0\ 1)$

while *stopping condition is not met* **do**

Compute $\nabla f(\mathbf{x}_k)$ and step size t

Compute $\alpha_{k+1} = \dfrac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}$, $\beta_k = \dfrac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$

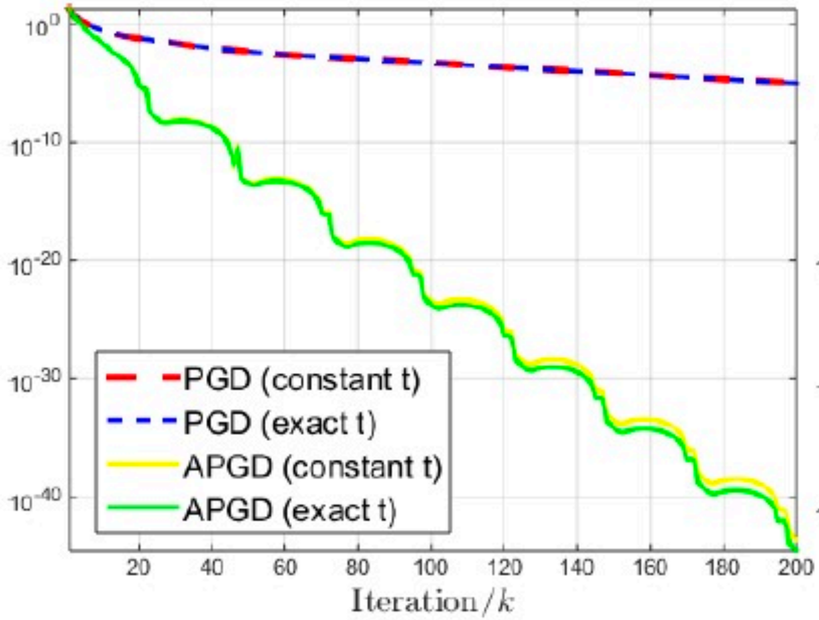
$\mathbf{y}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k)$

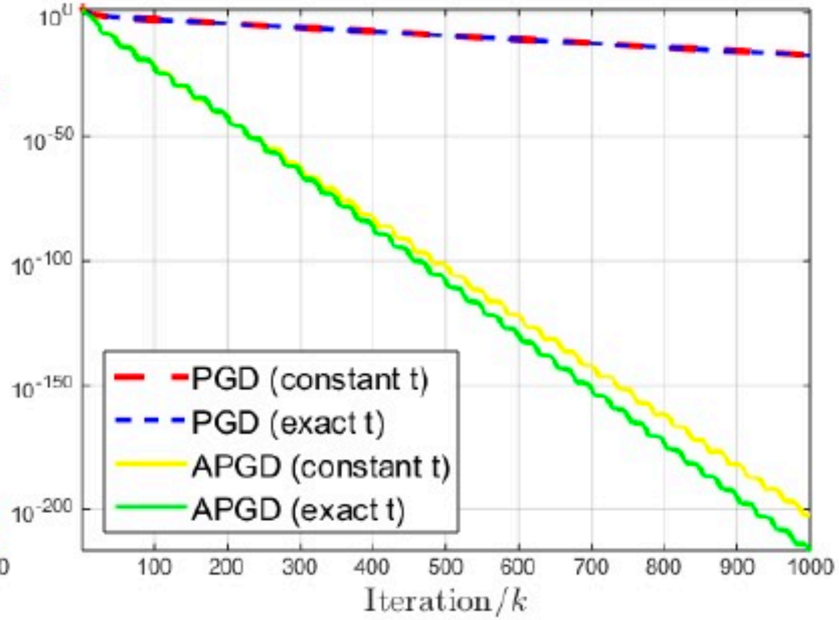
$\mathbf{x}_{k+1} = \mathbf{y}_{k+1} + \beta_k(\mathbf{y}_{k+1} - \mathbf{y}_k)$

end

Convergence rate

Recall the observation : different slope $\Longrightarrow$ different convergence rate





In general, for GD, the objective function decrease in the order of $\mathcal{O}\left(\frac{1}{k}\right)$.

But for Accelerated gradient, it drops in the order of $\mathcal{O}\left(\frac{1}{k^2}\right)$!

And it is *optimal*: you can never do better than $\mathcal{O}\left(\frac{1}{k^2}\right)$, if you only use gradient information!

Summary

(Recall) Special case: Quadratic Optimization (convex optimization)

- Sometime called Quadratic Programming (QP)

$$f(x) = \frac{1}{2} x^\top Q x - b^\top x$$

$$\nabla f(x) = Qx - b$$

$$Q = Q^\top > 0,\ b \in \mathbb{R}^n$$

Convex!!!

Unconstrained Minimization

$$\min_{x \in \mathbb{R}^n} f(x)$$

- Assumption
  - f is convex and continuously differentiable
  - The optimal value $f^* = \inf_{x \in \mathbb{R}^n} f(x)$ is finite
- Minimization methods
  - Iterative methods for the form

$$x_{k+1} = x_k + \alpha_k d_k, \; x_0 = x \in \mathbb{R}^n$$

    α_k: step size
    d_k: direction of the iterative algorithm
  - Generate a sequence of points $\{x_k\}$ such that $f(x_k) \to f^*$ as $k \to \infty$
  - Can be interpreted as iterative methods for solving the system of equations to satisfy the necessary and sufficient optimality condition

$$\nabla f(x^*) = 0$$

Unconstrained Minimization: Gradient Descent Method

- Gradient descent algorithm

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \; x_0 = x$$

- α_k: stepsize
  - Constant
  - Diminishing: $\alpha_k \to 0$ with $\sum_{k=1}^{\infty} \alpha_k = \infty$
  - Linear search types (optimal stepsize, hard to find)
    - Exact line search: $\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha d_k)$
    - Backtracking line search

Steepest Gradient Descent for QP

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)}).$$

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\boldsymbol{x}^{(k)} - \alpha \nabla f(\boldsymbol{x}^{(k)})).$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \frac{\boldsymbol{g}^{(k)\top}\boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)\top}\boldsymbol{Q}\boldsymbol{g}^{(k)}}\boldsymbol{g}^{(k)},$$

$$\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)}) = \boldsymbol{Q}\boldsymbol{x}^{(k)} - \boldsymbol{b}.$$

Theorem 8.6 *Let $\{\boldsymbol{x}^{(k)}\}$ be a convergent sequence of iterates of the steepest descent algorithm applied to a function f. Then, the order of convergence of $\{\boldsymbol{x}^{(k)}\}$ is 1 in the worst case; that is, there exist a function f and an initial condition $\boldsymbol{x}^{(0)}$ such that the order of convergence of $\{\boldsymbol{x}^{(k)}\}$ is equal to 1.* $\square$

$\mathcal{O}\left(\frac{1}{k}\right)$ $$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\| = O(\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|^p),$$ p=1

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \frac{\boldsymbol{g}^{(k)\top}\boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)\top}\boldsymbol{Q}\boldsymbol{g}^{(k)}}\boldsymbol{g}^{(k)},$$

$$\boldsymbol{g}^{(k)} = \nabla f(\boldsymbol{x}^{(k)}) = \boldsymbol{Q}\boldsymbol{x}^{(k)} - \boldsymbol{b}.$$

Gradient Descent with Strong Convexity

Theorem

Suppose that f is strongly convex, i.e., $mI \leq \nabla^2 f(x) \leq MI$ for all x. Then with the constant stepsize $\alpha < \frac{\min(2,m)}{M}$,

$$\|x_k - x^*\| \leq cq^k \quad 0 < q < 1$$

That is, the gradient descent algorithm has the geometric convergence rate.

Using the strong convexity assumption,

$$\|x_{k+1} - x^*\|^2 \leq (1 - m\alpha + \alpha^2 M)^{k+1} \|x_0 - x^*\|^2$$

- Geometric convergence is not bad, but there are not many functions that satisfy the strong convexity assumption
- Note that with the strongly convexity, x^* is unique

Algorithm: Nesterov's accelerated gradient for $(\mathcal{P})$

Result: A solution $\mathbf{x}$ that approximately solves $(\mathcal{P})$

Initialization Set $\mathbf{x}_0 \in \mathbb{R}^n$ $\alpha_1 \in (0\ 1)$

while *stopping condition is not met* **do**

Compute $\nabla f(\mathbf{x}_k)$ and step size t

Compute $\alpha_{k+1} = \dfrac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}$, $\beta_k = \dfrac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$

$\mathbf{y}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k)$

$\mathbf{x}_{k+1} = \mathbf{y}_{k+1} + \beta_k(\mathbf{y}_{k+1} - \mathbf{y}_k)$

end