

Lecture 8 - Algorithms for unconstrained optimization - Gradient Descent

Dinesh Krishnamoorthy

Department of Engineering Cybernetics, NTNU

Spring 2026

Sources

Presentation based on material from:

- ▶ Nocedal & Wright (2006): Numerical optimization (Chapters 2-4)
- ▶ Boyd & Vandenberghe (2004): Convex optimization (Chapter 9)

Objectives and Readings

Learning Objectives:

1. *understand* algorithms for solving unconstrained optimization problem, including gradient descent, Newton, and quasi-Newton search for unconstrained minimization, and the line search and trust region variants of these algorithms.
2. *understand* the role of 1-variable minimization in line search based optimization, exact and inexact line search methods (sufficient decrease and backtracking)
3. *can demonstrate and apply* the understanding of these algorithms for small analytical optimization problems of a few variables.

Readings: Sects. 9.1,9.2, 9.3, 9.5 *Boyd & Vandenberghe (2004): Convex optimization*, 2.2, 3.1, 4.1 Nocedal & Wright (2006): Numerical optimization

Outline

1. Introduction
2. Local exploration: Descent methods
3. Line search methods

Unconstrained problem

$$\begin{array}{ll} \min_x & f(x) \\ \text{with} & x \in \mathbb{R}^n \end{array}$$

Smoothness assumption: $f(x)$ twice continuously differentiable.

We denote the optimal value

$$\inf_x f(x) =: f(x^*)$$

Existence

Weierstrass Theorem: Sufficient condition for the existence of properly defined optima

One dimensional ($x \in \mathbb{R}$)

- ▶ A continuous function $f(x)$ defined on a closed finite interval attains its maximum and minimum within that interval

Multi-dimensional ($x \in \mathbb{R}^n$)

- ▶ A function $f(x)$ that is continuous on a closed and bounded set in $\mathbb{R}^n$ attains its maximum and minimum in that set

Existence

Weierstrass Theorem: Sufficient condition for the existence of properly defined optima

In general

$$\min_x f(x) \geq \inf_x f(x)$$

If Weierstrass's theorem applies, then

$$\min_x f(x) = \inf_x f(x)$$

Recap: Optimality

First order necessity for unconstrained minimizer x^*

$$f(x^* + \delta x) = f(x^*) + \nabla f(x^*)^T \delta x + \frac{1}{2} \delta x^T H(x^*) \delta x + O(\|\delta x - x^*\|^2)$$

$$\delta f(x^*) = \nabla f(x^*)^T \delta x \geq 0 \quad \forall \delta x$$

$$\Rightarrow \quad \nabla f(x^*) = 0 \quad (\text{stationary point})$$

Second order sufficiency for stationary point x^*

$$\delta f(x^*) = \delta x^T H(x^*) \delta x > 0 \quad \forall \delta x$$

$$\Rightarrow \quad H(x^*) \text{ positive definite} \quad (\text{local minimizer})$$

Optimality

If f is differentiable and convex, a necessary and sufficient condition for a point x^* to be optimal is

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

Key idea

Solving unconstrained minimization problem is same as finding a solution to $\nabla f(x) = 0$ which is a set of n equations and n variables.

In a few special cases we can find the optimum by analytically solving the optimality condition $\nabla f(x) = 0$.

Example 1

$$f = (x_1 - 2)^2 + (x_2 - 3)^2$$

Necessary and sufficient conditions (since f is convex)

$$\left. \begin{aligned} \frac{\partial f}{\partial x_1} &= 2(x_1 - 2) = 0 \\ \frac{\partial f}{\partial x_2} &= 2(x_2 - 3) = 0 \end{aligned} \right\} \quad \text{stationary point } x^* = (2, 3)^T$$

Stationary point is a local minimizer $x^* = (2, 3)^T$

Example 2

$$f = 2x_1 + x_1^{-2} + 2x_2 + x_2^{-2}$$

Stationary conditions

$$\left. \begin{aligned} \frac{\partial f}{\partial x_1} &= 2 - 2x_1^{-3} = 0 \\ \frac{\partial f}{\partial x_2} &= 2 - 2x_2^{-3} = 0 \end{aligned} \right\} \quad \text{stationary point } \mathbf{x}^* = (1, 1)^T$$

Sufficient condition

$$H = \begin{bmatrix} 6x_1^{-4} & 0 \\ 0 & 6x_2^{-4} \end{bmatrix} \Rightarrow H(\mathbf{x}^*) \text{ positive definite}$$

Stationary point is a local minimizer $\mathbf{x}^* = (1, 1)^T$

Example 3

$$f = x_1 + x_2 + x_1 e^{-x_2} + x_2^2 e^{-x_1}$$

Stationary conditions

$$\begin{aligned}\frac{\partial f}{\partial x_1} &= 1 + e^{-x_2} - x_2^2 e^{-x_1} = 0 \\ \frac{\partial f}{\partial x_2} &= 1 - x_1 e^{-x_2} + 2x_2 e^{-x_1} = 0\end{aligned}$$

Cannot be solved explicitly

$\Rightarrow$ have to resort to an iterative procedure

Iterative algorithm

Sequence of points x_k (iterates): $x_0, x_1, x_2, \dots$

Minimizing algorithm: $x_{k+1} = A(x_k)$

Global convergence

- ▶ Ability of algorithm A to reach the neighborhood of a (local) minimizer x^* from an arbitrary initial point x_0 not close to x^* .
- ▶ Convergence of A should not be affected by choice of x_0 .

Local convergence

- ▶ Ability of algorithm A to approach x^* rapidly from a starting point x_0 or iterate x_k in the neighborhood of x^* .
- ▶ Local convergence rate can be measured by the decrease of error $x_k - x^*$ in subsequent iterations:

$\ x_{k+1} - x^*\ \leq \gamma \ x_k - x^*\ ,$	$0 < \gamma < 1$	Linear
$\ x_{k+1} - x^*\ \leq \gamma_k \ x_k - x^*\ ,$	$\gamma_k \rightarrow 0$	Superlinear
$\ x_{k+1} - x^*\ \leq \gamma \ x_k - x^*\ ^2,$	$\gamma_k \in \mathbb{R}_+$	Quadratic

What did you learn?

Unconstrained Optimization

$$\min_x f(x)$$

Necessary condition of optimality: $\nabla_x f(x^*) = 0$

Analytical solution for
a few special cases

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

n unknowns & n equations

Solve using an
iterative procedure

$$x_{k+1} = x_k + \alpha s_k$$

Choose step length

Choose search direction

Local exploration: Descent methods

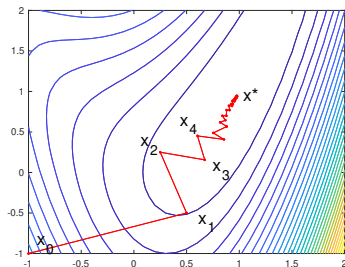
Iterative procedure:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{s}_k$$

where $\mathbf{s}_k$ is the search direction and α is the step size

Descent methods: $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$

for every iteration



Descent (downward) direction requires

$$\delta f = \nabla f(\mathbf{x}_k)^T \delta \mathbf{x} < 0 \quad \Rightarrow \quad \nabla f(\mathbf{x}_k)^T \mathbf{s}_k < 0$$

General descent method

Iterative procedure that alternates between determining descent direction and choosing a step size.

Given a starting point x , repeat

1. Determine a descent direction s_k
2. Choose a step size $\alpha > 0$ (Line search)
3. Update $x_{k+1} = x_k + \alpha s_k$

Steepest descent direction

Linear approximation

$$f_{k+1} = f_k + \nabla f(x_k)^T \delta x$$

motivates steepest descent direction

$$s_k = -\nabla f(x_k)$$

A step αs_k in direction s_k satisfies requirement

$$\delta f = -\nabla f(x_k)^T \alpha \nabla f(x_k) < 0$$

Gradient method

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

Gradient algorithm

1. For $k = 0$: select $x_0, \varepsilon > 0$
2. For $k \geq 0$:
 - ▶ set $s_k = -\nabla f(x_k)$
 - ▶ Choose step size $\alpha_k > 0$ (line search)
 - ▶ set $x_{k+1} = x_k + \alpha_k s_k$
3. Stop if $\|\nabla f(x_{k+1})\| < \varepsilon$, else set $k = k + 1$ and repeat step 2

Termination criteria

Condition on optimality

$$\|\nabla f(\mathbf{x}_{k+1})\| < \varepsilon$$

or a condition on change in $\mathbf{x}$

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$$

or a condition on change in f

$$(f_{k+1} - f_k) < -\varepsilon$$

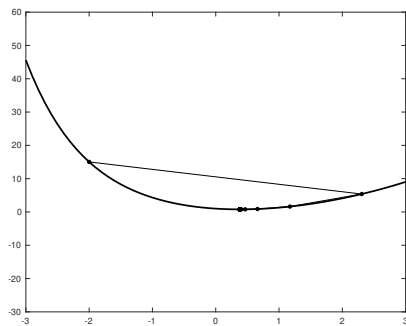
or a combination of these, with $\varepsilon > 0$,
possibly with an additional bound on the number of iterations

$$k < k_{max}$$

NB! Look at `exitflag` in MATLAB functions `fmincon`, `fminunc`, etc.

Example

$$\min_x x^2 + e^{-1.2x}$$

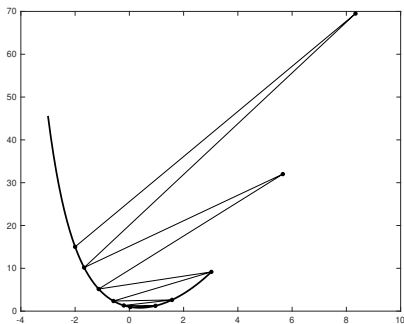


$$\alpha = 0.25$$

Example

$$\min_x x^2 + e^{-1.2x}$$

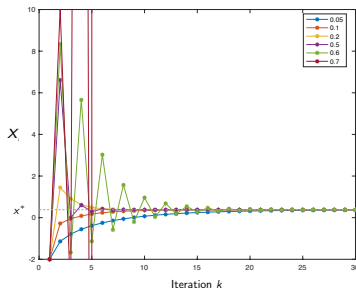
Gradient Descent for 10 iterations



$$\alpha = 0.6$$

Step length

- ▶ Choosing a small step length α would lead to very slow convergence
- ▶ Choosing a large step length α would lead to divergence



Choosing a good step length α is an important practical problem.

Exact line search

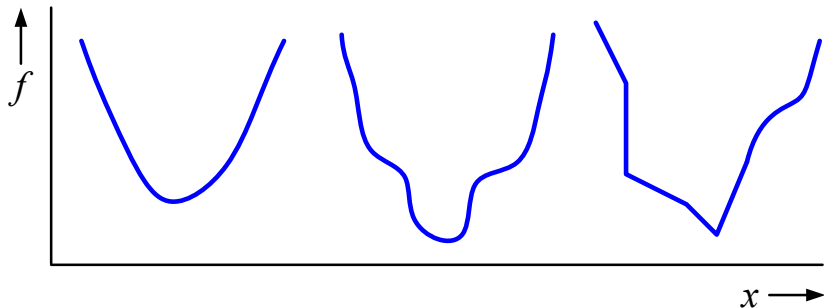
Choose α that minimizes f along the ray $\{x + \alpha s \mid \alpha \geq 0\}$

$$\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x_k + \alpha s_k)$$

Line search is a one-dimensional minimization problem in α
with

- s_k : search direction vector
- α : search step
- x_k : starting point of iteration k

One variable minimization: unimodality

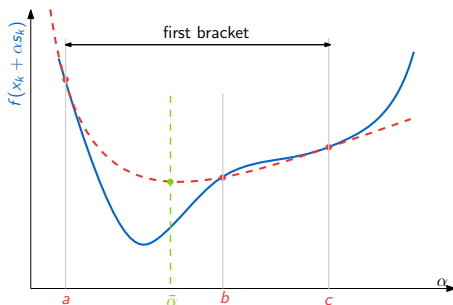


Unimodal function:

- ▶ unique minimum
- ▶ not necessarily continuous/differentiable
- ▶ monotonically decreasing and then increasing

One variable minimization: quadratic interpolation

1. Search along the line for three consecutive points $a < b < c$, such that $f(a) > f(b) < f(c)$ (high-low-high).
2. Fit a quadratic function from the three points and find minimum $\bar{\alpha}$. Note: $\bar{\alpha} \in (a, c)$.
3. Replace a or c with $\bar{\alpha}$, such that the new points form high-low-high, and repeat.



Example

Consider the function:

$$f = 1 - \frac{2x}{3^x}$$

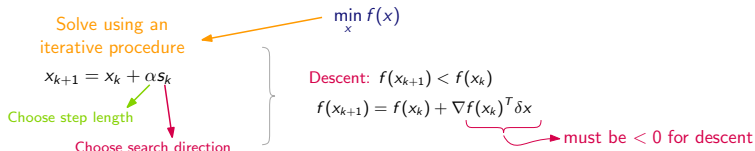
We apply a quadratic interpolation procedure to minimize f .

The initial bracket is set to $[0, 2]$, the third point to $x = 1$. Determine the next bracket.

Solution: See handwritten notes.

Exact Line search

Unconstrained Optimization



Gradient descent algorithm

1. Choose $s_k = -\nabla f(x_k)$
 2. Choose step length α
 3. Update $x_{k+1} = x_k + \alpha s_k$
- Exact line search
- $$\alpha = \arg \min_{\alpha \geq 0} f(x_k + \alpha s_k)$$
-

Exact line search

Line search is a one-dimensional minimization problem in α
with

- s_k : search direction vector
- α : search step
- x_k : starting point of iteration k

Exact line search is used if cost of this one-variable minimization is low compared to the cost of computing the search direction itself.

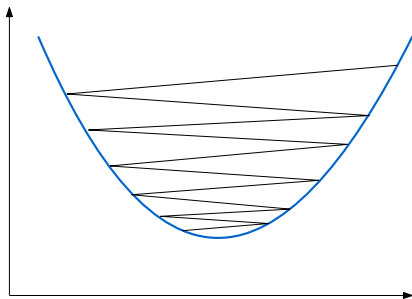
- ▶ Exact line search may be expensive, especially if f is not convex
- ▶ Finding optimal α_k is also often unnecessary

Line search: Inexact solution

A simple condition we could impose when choosing α is reduction in f :

$$f(x_k + \alpha s_k) < f(x_k)$$

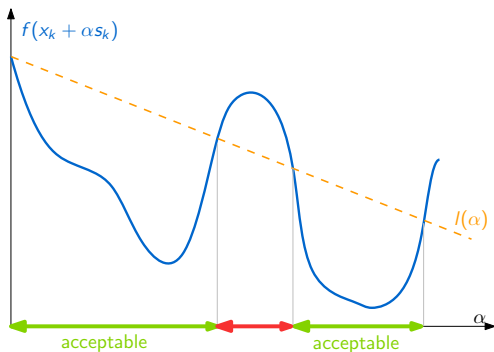
But this is insufficient to produce convergence to x^*



Line search: Inexact solution

Armijo condition: α_k should give sufficient decrease.

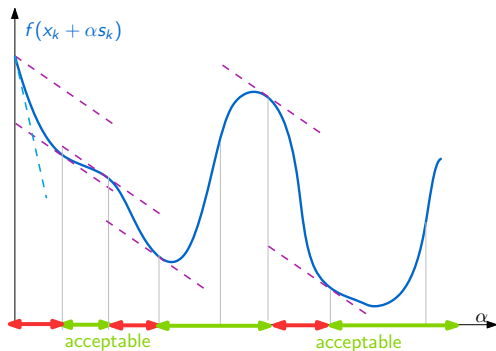
$$f(x_k + \alpha s_k) < f(x_k) + \underbrace{c_1 \alpha \nabla f(x_k)^\top s_k}_{:=l(\alpha)}, \quad c_1 \in (0, 1)$$



Line search: Inexact solution

To rule out unacceptably short steps, we need a curvature condition:

$$\nabla f(x_k + \alpha s_k)^T s_k \geq c_2 \nabla f(x_k)^T s_k, \quad c_2 \in (c_1, 1)$$

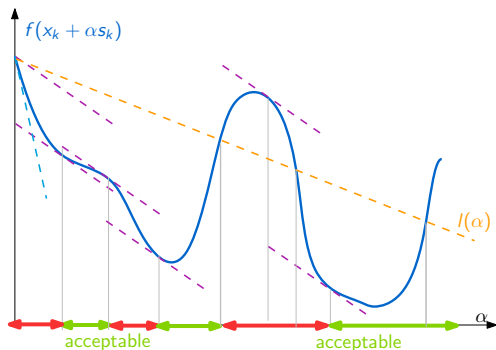


Line search: Inexact solution

Wolfe condition: Armijo + curvature

$$f(x_k + \alpha s_k) < f(x_k) + c_1 \alpha \nabla f(x_k)^T s_k$$

$$|\nabla f(x_k + \alpha s_k)^T s_k| \leq c_2 |\nabla f(x_k)^T s_k|, \quad 0 < c_1 < c_2 < 1$$



Typical values for Wolfe: $c_1 = 10^{-4}$ $c_2 = 0.9$.

Inexact line search by backtracking

Alternatively, starting from a value close to 1, if the line search algorithm chooses α appropriately by using a backtracking approach, then Armijo condition (sufficient decrease) is enough to terminate the line search procedure.

1. Choose $\bar{\alpha} = 1$, $c_1 \in (0, 1)$, $\rho \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$
2. Evaluate $f(x_k + \alpha s_k)$
3. If Armijo condition not satisfied, set $\alpha \leftarrow \rho\alpha$, and return to step 2.
4. Terminate with $\alpha_k = \alpha$

(Nocedal and Wright, 2006)

Example

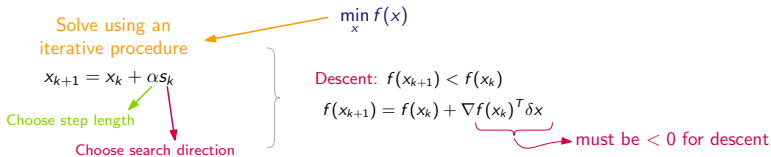
Consider the function:

$$f = (1 - x_1)^2 + (x_2 - x_1^2)^2$$

We apply the gradient descent algorithm with backtracking inexact line search using Armijo's sufficient reduction criterion. Compute *two* iterates starting from $\mathbf{x}_0 = (0, 0)^T$. Use $\bar{\alpha} = 1$, $\rho = 0.5$, and $c_1 = 0.1$.

Solution: See handwritten notes.

Unconstrained Optimization



Gradient descent algorithm

1. Choose $s_k = -\nabla f(x_k)$

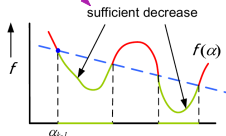
2. Choose step length α

3. Update $x_{k+1} = x_k + \alpha s_k$

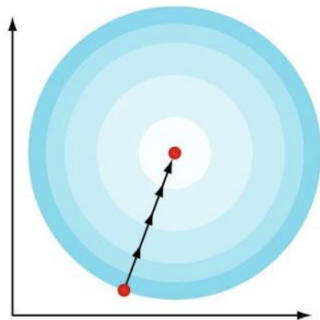
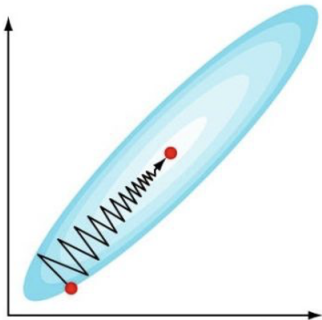
Exact line search

$$\alpha = \arg \min_{\alpha \geq 0} f(x_k + \alpha s_k)$$

Backtracking line search



Gradient method

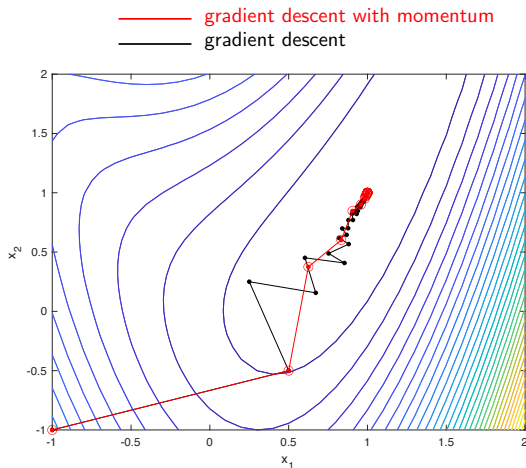


Gradient method

- ▶ exhibits approximately linear convergence
- ▶ convergence rate depends on the condition number of the Hessian near x^*
- ▶ scaling is important
- ▶ convergence can be very slow
- ▶ main advantage is it's simplicity

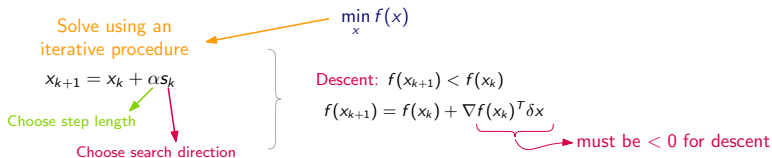
Steepest descent with extrapolation

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



Summary

Unconstrained Optimization



Gradient descent algorithm

1. Choose $s_k = -\nabla f(x_k)$
 2. Choose step length α
 3. Update $x_{k+1} = x_k + \alpha s_k$
- Exact line search
- $$\alpha = \arg \min_{\alpha \geq 0} f(x_k + \alpha s_k)$$
- Backtracking line search

