Statistical Optimization: Lecture 8

Line Search Methods: Gradient Descent Step Size Selection

Zijian Guo

Zhejiang University
Center for Data Science

April 6, 2026

Step size in gradient descent

For gradient descent, the update takes the form

$$\theta_{t+1} = \theta_t - \gamma_t \nabla f(\theta_t).$$

In theoretical analysis, one often chooses a fixed step size, for example

$$\gamma_t \equiv \frac{1}{L},$$

when f is L-smooth.

This choice is simple, guarantees descent, and is convenient for proving convergence.

Objective

In practice, however, a fixed step size may not be the best choice.

If the step size is too small:

- the method is stable,
- but progress can be very slow.

If the step size is too large:

- the iterates may oscillate,
- or even fail to decrease the objective.

So it is natural to ask: *how should we choose the step size at each iteration?*

Outline

Line search

Wolfe conditions

Backtracking method

Case study: Logistic regression

Line search setup

To choose the step size adaptively, we consider the more general update

$$\theta_{t+1} = \theta_t + \gamma_t p_t,$$

where p_t is a search direction and $\gamma_t > 0$ is the step size.

In line search methods, we usually choose p_t to be a descent direction, that is,

$$\nabla f(\theta_t)^\top p_t < 0,$$

so that a small positive step along p_t decreases the objective.

For gradient descent, we take

$$p_t = -\nabla f(\theta_t).$$

Line search setup

Then define the one-dimensional function

$$\phi(\gamma_t) := f(\theta_t + \gamma_t \boldsymbol{p}_t), \qquad \gamma_t > 0.$$

Thus, each iteration reduces the multidimensional problem to a one-dimensional search along the direction $\boldsymbol{p}_t$.

Exact line search

Once the direction p_t is chosen, the step length γ_t can be obtained by considering

$$\min_{\gamma_t > 0} \phi(\gamma_t) \;=\; \min_{\gamma_t > 0} f(\theta_t + \gamma_t p_t).$$

If we solve this one-dimensional problem exactly, we derive the maximum benefit from the direction p_t.

In practice, this may be done by bisection search, golden-section search and other methods.

Exact line search

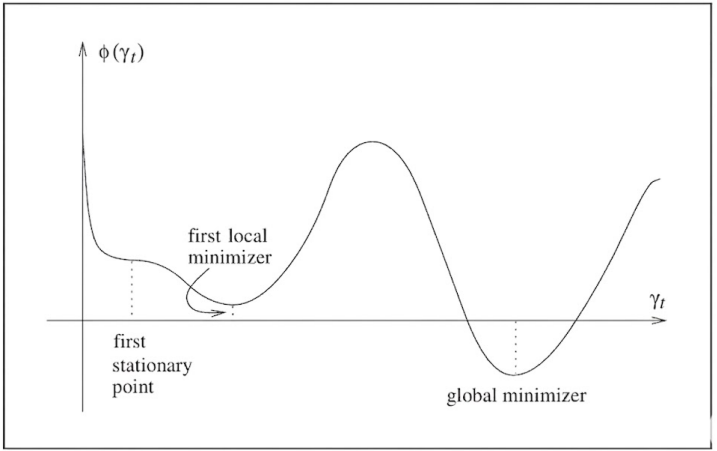

Figure: The ideal step length is the global minimizer.

Inexact line search

However, exact minimization may be expensive and is usually unnecessary.

In practice, we only seek a step length that gives a sufficient decrease of the objective and has reasonable practical cost.

So the basic question in line search is:

How can we define conditions that make a step length acceptable?

This leads to the termination conditions for line search, starting with the Wolfe conditions.

Outline

Line search

Wolfe conditions

Backtracking method

Case study: Logistic regression

Sufficient decrease condition

A popular inexact line search condition first requires sufficient decrease:

$$f(\theta_t + \gamma_t p_t) \leq f(\theta_t) + c_1 \gamma_t \nabla f(\theta_t)^\top p_t,$$

for some constant $c_1 \in (0, 1)$.

This inequality is sometimes called Armijo condition.

Remark.

- The reduction in f should be proportional to both the step length γ_t and the directional derivative $\nabla f(\theta_t)^\top p_t$.
- The right-hand side is a linear function of γ_t; denote it by $l(\gamma_t)$.
- This condition alone is not enough, since it is satisfied for all sufficiently small γ_t.

Sufficient decrease condition

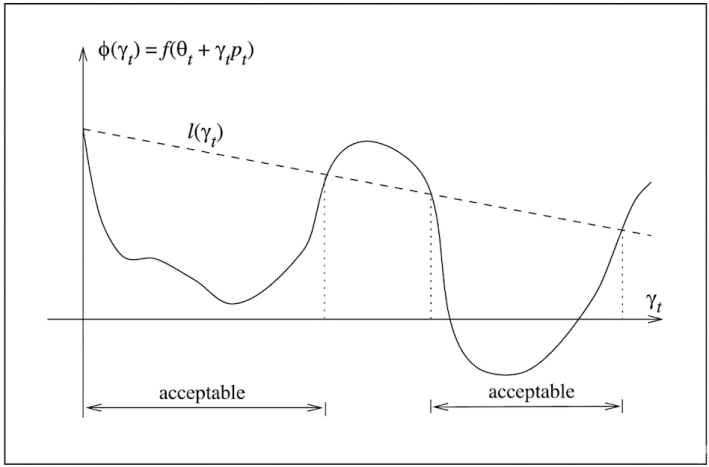

Figure: Sufficient decrease condition.

Curvature condition

To rule out unacceptably short steps, we introduce a second requirement, called the curvature condition:

$$\nabla f(\theta_t + \gamma_t p_t)^\top p_t \geq c_2 \nabla f(\theta_t)^\top p_t,$$

for some constant $c_2 \in (c_1, 1)$.

Since

$$\phi'(\gamma_t) = \nabla f(\theta_t + \gamma_t p_t)^\top p_t,$$

and p_t is a descent direction, we have $\phi'(0) = \nabla f(\theta_t)^\top p_t < 0$. The curvature condition requires us to look for a "not so fast descent"

$$\phi'(\gamma_t) \geq c_2 \phi'(0).$$

Curvature condition

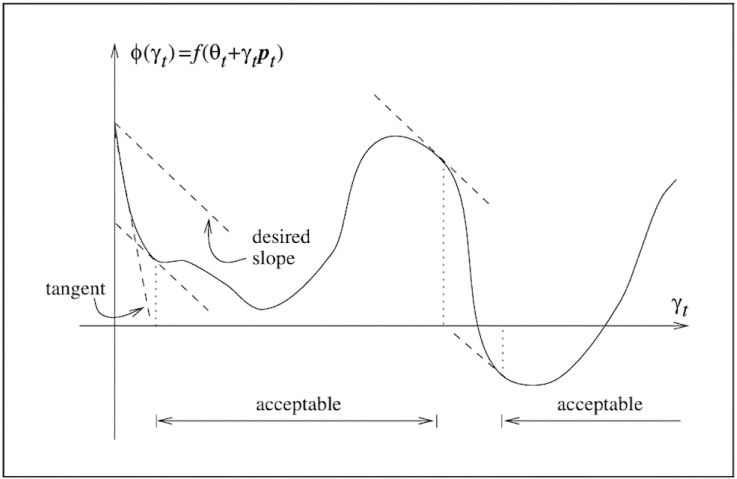

Figure: The curvature condition.

Wolfe conditions

Wolfe conditions. Together, the sufficient decrease and curvature conditions are called the Wolfe conditions:

$$\begin{cases} f(\theta_t + \gamma_t p_t) \leq f(\theta_t) + c_1 \gamma_t \nabla f(\theta_t)^\top p_t, \\ \nabla f(\theta_t + \gamma_t p_t)^\top p_t \geq c_2 \nabla f(\theta_t)^\top p_t, \end{cases} \qquad 0 < c_1 < c_2 < 1.$$

The first condition enforces sufficient decrease, while the second prevents the step length from being too short.

Wolfe conditions

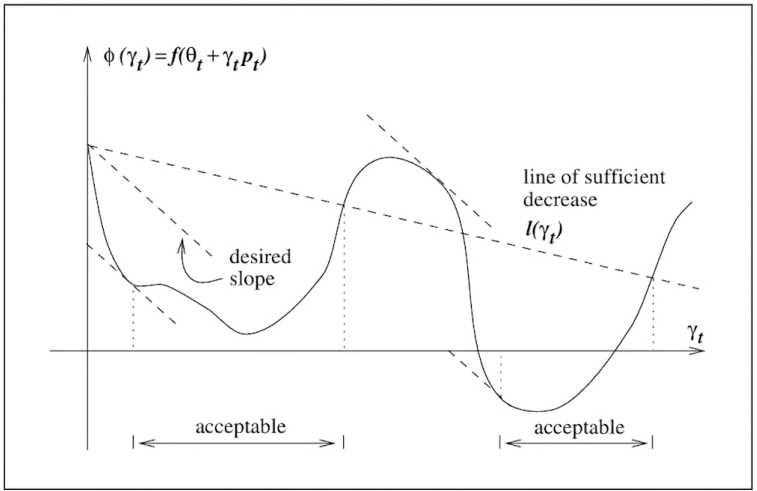

Figure: Step lengths satisfying the Wolfe conditions.

Strong Wolfe conditions

A commonly used variant is the strong Wolfe conditions:

$$\begin{cases} f(\theta_t + \gamma_t p_t) \leq f(\theta_t) + c_1 \gamma_t \nabla f(\theta_t)^\top p_t, \\ \left| \nabla f(\theta_t + \gamma_t p_t)^\top p_t \right| \leq c_2 \left| \nabla f(\theta_t)^\top p_t \right|, \end{cases} \qquad 0 < c_1 < c_2 < 1.$$

The strong Wolfe conditions require not only that the slope should not remain too negative, but also that even if it becomes positive, it should not be too large.

Equivalently, the directional derivative at the new point must be small in magnitude.

Outline

Line search

Wolfe conditions

Backtracking method

Case study: Logistic regression

Sufficient decrease and backtracking

If p_t is a descent direction, then

$$\nabla f(\theta_t)^\top p_t < 0,$$

and the Armijo condition

$$f(\theta_t + \gamma_t p_t) \leq f(\theta_t) + c_1 \gamma_t \nabla f(\theta_t)^\top p_t$$

is automatically satisfied for all sufficiently small $\gamma_t > 0$.
Backtracking: Starting from an initial trial step length, we repeatedly shorten it until the sufficient decrease condition is satisfied.

Backtracking line search

Algorithm

Input: current iterate θ_t, search direction p_t, initial trial step length $\bar{\gamma} > 0$
Parameters: Shrinking parameter $\rho \in (0, 1)$, $c_1 \in (0, 1)$

1. Set $\gamma \leftarrow \bar{\gamma}$.
2. While
$$f(\theta_t + \gamma p_t) > f(\theta_t) + c_1 \gamma \nabla f(\theta_t)^\top p_t,$$
set
$$\gamma \leftarrow \rho\gamma.$$
3. Return $\gamma_t = \gamma$.

Similar ideas can also be used in line search procedures based on the Wolfe conditions.

Outline

Line search

Wolfe conditions

Backtracking method

Case study: Logistic regression

Case study: logistic regression

We consider binary classification with logistic regression.

Given data (x_i, y_i) with $y_i \in \{0, 1\}$, we minimize the empirical logistic loss

$$f(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\Big[y_i \log \sigma(x_i^\top \theta) + (1 - y_i)\log\big(1 - \sigma(x_i^\top \theta)\big)\Big], \quad \text{with} \quad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

The gradient of the logistic loss is

$$\nabla f(\theta) = \frac{1}{n}\sum_{i=1}^{n}\big(\sigma(x_i^\top \theta) - y_i\big)x_i.$$

Equivalently, in matrix form,

$$\nabla f(\theta) = \frac{1}{n}X^\top(\hat{p} - y), \qquad \hat{p}_i = \sigma(x_i^\top \theta),\ i = 1, \ldots, n.$$

Gradient descent with Wolfe rules line search

Given an initial point θ_0, tolerance $\mathrm{tol} > 0$, and maximum iteration number $\mathrm{max\_iter}$:

1. For $t = 0, 1, 2, \ldots, \mathrm{max\_iter} - 1$:
   - 1.1 Compute the gradient
   $$g_t = \nabla f(\theta_t).$$
   - 1.2 **Stopping criterion:** if
   $$\|g_t\|_2 \leq \mathrm{tol},$$
   stop the algorithm.
   - 1.3 Choose the search direction
   $$p_t = -g_t.$$
   - 1.4 Choose the step size γ_t by Wolfe-style backtracking line search.
   - 1.5 Update
   $$\theta_{t+1} = \theta_t + \gamma_t p_t.$$

Wolfe-style backtracking line search

Given the current iterate θ_t and direction $p_t = -\nabla f(\theta_t)$:

1. Initialize $\gamma \leftarrow \bar{\gamma}$.
2. While either of the following conditions fails,

$$f(\theta_t + \gamma p_t) \leq f(\theta_t) + c_1 \gamma \nabla f(\theta_t)^\top p_t,$$

$$\nabla f(\theta_t + \gamma p_t)^\top p_t \geq c_2 \nabla f(\theta_t)^\top p_t,$$

   shrink the step:

$$\gamma \leftarrow \rho\gamma.$$

3. Return $\gamma_t = \gamma$

Results Compare

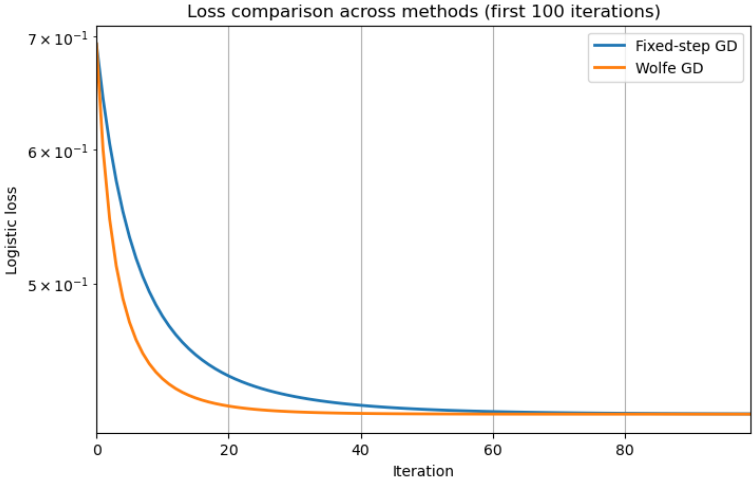

Figure: Wolfe GD vs. Fixed-step GD