Optimization for Machine Learning
机器学习中的优化方法

陈 程

华东师范大学 软件工程学院

chchen@sei.ecnu.edu.cn

Review of first-order methods

condition	Subgradient descent	proximal GD	Nestrov's AGD	Lower bound
convex	$O\left(\frac{1}{\varepsilon^2}\right)$	$O\left(\frac{1}{\varepsilon}\right)$	$O(\frac{1}{\sqrt{\varepsilon}})$	$\Omega(\frac{1}{\sqrt{\varepsilon}})$
strongly convex	$O\left(\frac{1}{\varepsilon}\right)$	$O\left(\kappa \log \frac{1}{\varepsilon}\right)$	$O\left(\sqrt{\kappa} \log \frac{1}{\varepsilon}\right)$	$\Omega\left(\sqrt{\kappa} \log \frac{1}{\varepsilon}\right)$

Table: Iteration complexity of first-order methods

Outline

Newton's methods

Recall that optimizing smooth function $f(\mathbf{x})$ by gradient descent

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1}{L}\nabla f(\mathbf{x}_t)$$

is achieved by minimizing

$$\min_{\mathbf{x}} f(\mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle + \frac{L}{2}\left\| \mathbf{x} - \mathbf{x}_t \right\|_2^2.$$

If we can compute Hessian matrix, we can minimize

$$\min_{\mathbf{x}} f(\mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle + \frac{1}{2}\langle \mathbf{x} - \mathbf{x}_t, \nabla^2 f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t) \rangle.$$

Suppose $\nabla^2 f(\mathbf{x}_t)$ is non-singular, then we achieve Newton's method

$$\mathbf{x}_{t+1} = \mathbf{x}_t - (\nabla^2 f(\mathbf{x}_t))^{-1}\nabla f(\mathbf{x}_t).$$

Local quadratic convergence rate

Suppose the twice differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ has L_2-Lipschitz continuous Hessian and local minimizer $\mathbf{x}^*$ with $\nabla^2 f(\mathbf{x}^*) \succeq \mu \mathbf{I}$, then the Newton's method

$$\mathbf{x}_{t+1} = \mathbf{x}_t - (\nabla^2 f(\mathbf{x}_t))^{-1} \nabla f(\mathbf{x}_t)$$

with $\|\mathbf{x}_0 - \mathbf{x}^*\|_2 \le \mu/(2L_2)$ holds that

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2 \le \frac{L_2}{\mu} \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 .$$

Local quadratic convergence rate

The quadratic convergence means

$$\frac{L_2}{\mu} \left\|\mathbf{x}_{t+1} - \mathbf{x}^*\right\|_2 \leq \left(\frac{L_2}{\mu} \left\|\mathbf{x}_t - \mathbf{x}^*\right\|_2\right)^2$$

which leads to

$$\frac{L_2}{\mu} \left\|\mathbf{x}_T - \mathbf{x}^*\right\|_2 \leq \left(\frac{L_2}{\mu} \left\|\mathbf{x}_0 - \mathbf{x}^*\right\|_2\right)^{2^T}$$

The iteration complexity of Newton's method is $\mathcal{O}(\log\log(1/\epsilon))$.

Standard Newton's Method

Strengths:

1. The quadratic convergence is very fast (even for ill-conditioned case).

Weakness:

1. The convergence guarantee is local.
2. Each iteration requires $O(d^3)$ time.

Outline

Key idea

Approximate the Hessian matrix using only gradient information

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{G}_t^{-1} \nabla f(\mathbf{x}_t).$$

We hope:

- using only gradient information
- using limited memory
- achieving super-linear convergence

Secant equation

For quadratic function

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

we have $\nabla Q(\mathbf{x}_{t+1}) - \nabla Q(\mathbf{x}_t) = \nabla^2 Q(\mathbf{x}_{t+1})(\mathbf{x}_{t+1} - \mathbf{x}_t)$.

For general $f(\mathbf{x})$ with Lipschitz continuous Hessian, we have

$$\nabla f(\mathbf{x}_{t+1}) - \nabla f(\mathbf{x}_t) = \nabla^2 f(\mathbf{x}_{t+1})(\mathbf{x}_{t+1} - \mathbf{x}_t) + o(\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2),$$

which leads to

$$\nabla f(\mathbf{x}_{t+1}) - \nabla f(\mathbf{x}_t) \approx \nabla^2 f(\mathbf{x}_{t+1})(\mathbf{x}_{t+1} - \mathbf{x}_t).$$

Classical Quasi-Newton methods

Classical Quasi-Newton methods target to find $\mathbf{G}_{t+1}$ such that

$$\nabla f(\mathbf{x}_{t+1}) - \nabla f(\mathbf{x}_t) = \mathbf{G}_{t+1}(\mathbf{x}_{t+1} - \mathbf{x}_t)$$

and update the variable as

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{G}_t^{-1} \nabla f(\mathbf{x}_t).$$

The secant equation admit an infinite number of solutions. How to choose $\mathbf{G}_{t+1}$?

For given $\mathbf{G}_t$ or $\mathbf{G}_t^{-1}$, we hope

- $\{\mathbf{x}_t\}$ converges to $\mathbf{x}^*$ efficiently;
- $\mathbf{G}_{t+1}$ is close to $\mathbf{G}_t$;
- $\mathbf{G}_{t+1}$ or $\mathbf{G}_{t+1}^{-1}$ can be constructed/stored efficiently.

Woodbury matrix identity

The Woodbury matrix identity is

$$(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1},$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$, $\mathbf{C} \in \mathbb{R}^{k\times k}$, $\mathbf{U} \in \mathbb{R}^{d\times k}$ and $\mathbf{V} \in \mathbb{R}^{k\times d}$.

For $\mathbf{A} = \mathbf{G}_t$, $\mathbf{U} = \mathbf{Z}_t$, $\mathbf{V} = \mathbf{Z}_t^\top$ and $\mathbf{C} = \mathbf{I}$, we let

$$\mathbf{G}_{t+1} = \mathbf{G}_t + \mathbf{Z}_t\mathbf{Z}_t^\top,$$

then

$$\mathbf{G}_{t+1}^{-1} = \mathbf{G}_t^{-1} - \mathbf{G}_t^{-1}\mathbf{Z}_t(\mathbf{I} + \mathbf{Z}_t^\top\mathbf{G}_t^{-1}\mathbf{Z}_t)^{-1}\mathbf{Z}_t^\top\mathbf{G}_t^{-1}$$

can be computed within $\mathcal{O}(kd^2)$ flops for given $\mathbf{G}_t^{-1}$.

The SR1 method

We consider secant condition and the symmetric rank one (SR1) update

$$\begin{cases} \mathbf{y}_t = \mathbf{G}_{t+1}\mathbf{s}_t, \\ \mathbf{G}_{t+1} = \mathbf{G}_t + \mathbf{z}_t\mathbf{z}_t^\top. \end{cases}$$

where $\mathbf{s}_t = \mathbf{x}_{t+1} - \mathbf{x}_t$ and $\mathbf{y}_t = \nabla f(\mathbf{x}_{t+1}) - \nabla f(\mathbf{x}_t)$.

It implies

$$\mathbf{G}_{t+1} = \mathbf{G}_t + \frac{(\mathbf{y}_t - \mathbf{G}_t\mathbf{s}_t)(\mathbf{y}_t - \mathbf{G}_t\mathbf{s}_t)^\top}{(\mathbf{y}_t - \mathbf{G}_t\mathbf{s}_t)^\top\mathbf{s}_t}.$$

By Woodbury matrix identity, we have

$$\mathbf{G}_{t+1}^{-1} = \mathbf{G}_t^{-1} + \frac{(\mathbf{s}_t - \mathbf{G}_t^{-1}\mathbf{y}_t)(\mathbf{s}_t - \mathbf{G}_t^{-1}\mathbf{y}_t)^\top}{(\mathbf{s}_t - \mathbf{G}_t^{-1}\mathbf{y}_t)^\top\mathbf{y}_t}.$$

The updating time is $O(d^2)$ per iteration.

The Davidon-Fletcher-Powell (DFP) method

Let $\mathbf{G}_{t+1}$ be the solution of following matrix optimization problem

$$\min_{\mathbf{G}\in\mathbb{R}^{d\times d}} \|\mathbf{G}-\mathbf{G}_t\|_{\bar{\mathbf{G}}_t^{-1}}$$
$$\text{s.t} \quad \mathbf{G}=\mathbf{G}^\top, \;\; \mathbf{G}\mathbf{s}_t=\mathbf{y}_t,$$

where the weighted norm $\|\cdot\|_{\bar{\mathbf{G}}_t}$ is defined as

$$\|\mathbf{A}\|_{\bar{\mathbf{G}}_t} = \left\|\bar{\mathbf{G}}_t^{-1/2}\mathbf{A}\bar{\mathbf{G}}_t^{-1/2}\right\|_F \quad \text{with} \quad \bar{\mathbf{G}}_t = \int_0^1 \nabla^2 f(\mathbf{x}_t+\tau(\mathbf{x}_{t+1}-\mathbf{x}_t))\,\mathrm{d}\tau.$$

It implies DFP update

$$\mathbf{G}_{t+1} = \left(\mathbf{I}-\frac{\mathbf{y}_t\mathbf{s}_t^\top}{\mathbf{y}_t^\top\mathbf{s}_t}\right)\mathbf{G}_t\left(\mathbf{I}-\frac{\mathbf{s}_t\mathbf{y}_t^\top}{\mathbf{y}_t^\top\mathbf{s}_t}\right)+\frac{\mathbf{y}_t\mathbf{y}_t^\top}{\mathbf{y}_t^\top\mathbf{s}_t}.$$

The corresponding update to Hessian estimator is

$$\mathbf{G}_{t+1}^{-1} = \mathbf{G}_t^{-1} - \frac{\mathbf{G}_t^{-1}\mathbf{y}_t\mathbf{y}_t^\top\mathbf{G}_t^{-1}}{\mathbf{y}_t^\top\mathbf{G}_t^{-1}\mathbf{y}_t} + \frac{\mathbf{s}_t\mathbf{s}_t^\top}{\mathbf{y}_t^\top\mathbf{s}_t}. \quad \text{rank-2 update}$$

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method

Let $\mathbf{G}_{t+1}^{-1}$ be the solution of the following matrix optimization problem

$$\min_{\mathbf{H}\in\mathbb{R}^{d\times d}} \|\mathbf{H}-\mathbf{H}_t\|_{\bar{\mathbf{G}}_t}$$
$$\text{s.t} \quad \mathbf{H}=\mathbf{H}^\top, \;\; \mathbf{H}\mathbf{y}_t=\mathbf{s}_t,$$

where $\mathbf{H}_t=\mathbf{G}_t^{-1}$ and the weighted norm $\|\cdot\|_{\bar{\mathbf{G}}_t}$ is defined as

$$\|\mathbf{A}\|_{\bar{\mathbf{G}}_t}=\left\|\bar{\mathbf{G}}_t^{1/2}\mathbf{A}\bar{\mathbf{G}}_t^{1/2}\right\|_F \quad \text{with} \quad \bar{\mathbf{G}}_t=\int_0^1 \nabla^2 f(\mathbf{x}_t+\tau(\mathbf{x}_{t+1}-\mathbf{x}_t))\,\mathrm{d}\tau.$$

It implies BFGS update

$$\mathbf{G}_{t+1}^{-1}=\left(\mathbf{I}-\frac{\mathbf{s}_t\mathbf{y}_t^\top}{\mathbf{y}_t^\top\mathbf{s}_t}\right)\mathbf{G}_t^{-1}\left(\mathbf{I}-\frac{\mathbf{y}_t\mathbf{s}_t^\top}{\mathbf{y}_t^\top\mathbf{s}_t}\right)+\frac{\mathbf{s}_t\mathbf{s}_t^\top}{\mathbf{y}_t^\top\mathbf{s}_t}. \quad \text{rank-2 update}$$

The corresponding update to Hessian estimator is

$$\mathbf{G}_{t+1}=\mathbf{G}_t-\frac{\mathbf{G}_t\mathbf{s}_t\mathbf{s}_t^\top\mathbf{G}_t}{\mathbf{s}_t^\top\mathbf{G}_t\mathbf{s}_t}+\frac{\mathbf{y}_t\mathbf{y}_t^\top}{\mathbf{y}_t^\top\mathbf{s}_t}.$$

Local superlinear convergence

Theorem (informal)

Suppose f is strongly convex and has Lipschitz-continuous Hessian. Under mild conditions, SR1/DFP/BFGS achieves

$$\lim_{t\to\infty} \frac{\|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2}{\|\mathbf{x}_t - \mathbf{x}^*\|_2} = 0$$

- iteration complexity: larger than Newton methods but smaller than gradient methods
- asymptotic result: holds when $t \to \infty$

Explicit local convergence rate

Suppose the objective is μ-strongly-convex and L-smooth and let

$$\kappa = L/\mu \quad \text{and} \quad \lambda_t = \sqrt{\nabla f(\mathbf{x}_t)^\top (\nabla^2 f(\mathbf{x}_t))^{-1} \nabla f(\mathbf{x}_t)}.$$

1. For classical DFP method, we have

$$\lambda_t \leq \mathcal{O}\left(\left(\frac{\kappa^2 d}{t}\right)^{t/2}\right).$$

2. For classical BFGS method, we have

$$\lambda_t \leq \mathcal{O}\left(\left(\frac{\kappa d}{t}\right)^{t/2}\right).$$

3. For classical SR1 method, we have

$$\lambda_t \leq \mathcal{O}\left(\left(\frac{d \ln \kappa}{t}\right)^{t/2}\right).$$

Outline

Quasi-Newton Methods

Classical quasi-Newton methods are too expensive for large d.

- Each iteration requires $\mathcal{O}(d^2)$ time complexity.
- The space complexity is $\mathcal{O}(d^2)$.

Limited-memory BFGS (L-BFGS)

The BFGS update can be written as

$$\mathbf{H}_{t+1} = \mathbf{V}_t^\top \mathbf{H}_t \mathbf{V}_t + \rho_t \mathbf{s}_t \mathbf{s}_t^\top,$$

where $\rho_t = (\mathbf{y}_t^\top \mathbf{s}_t)^{-1}$ and $\mathbf{V}_t = \mathbf{I} - \rho_t \mathbf{y}_t \mathbf{s}_t^\top$.

Limited-memory BFGS method keeps the m most recent vector pairs

$$\{\mathbf{s}_i, \mathbf{y}_i\}_{i=k-m}^{k-1}$$

and applying BFGS update m times on some initial estimator $\mathbf{H}_{k,0} = \delta_{k,0}\mathbf{I}$.

Limited-memory BFGS (L-BFGS)

The update of L-BFGS can be written as

$$
\begin{aligned}
\mathbf{H}_k =& (\mathbf{V}_{k-1}^\top \dots \mathbf{V}_{k-m}^\top)\mathbf{H}_{k,0}(\mathbf{V}_{k-m} \dots \mathbf{V}_{k-1}) \\
&+ \rho_{k-m}(\mathbf{V}_{k-1}^\top \dots \mathbf{V}_{k-m+1}^\top)\mathbf{s}_{k-m}\mathbf{s}_{k-m}^\top(\mathbf{V}_{k-m+1} \dots \mathbf{V}_{k-1}) \\
&+ \rho_{k-m+1}(\mathbf{V}_{k-1}^\top \dots \mathbf{V}_{k-m+2}^\top)\mathbf{s}_{k-m+1}\mathbf{s}_{k-m+1}^\top(\mathbf{V}_{k-m+2} \dots \mathbf{V}_{k-1}) \\
&+ \dots \\
&+ \rho_{k-1}\mathbf{s}_{k-1}\mathbf{s}_{k-1}^\top.
\end{aligned}
$$

The iteration of L-BFGS is efficient for small m.

- Computing $\mathbf{H}_k \nabla f(\mathbf{x}_k)$ requires $\mathcal{O}(md)$ flops for given $\nabla f(\mathbf{x}_k)$.
- The storage of $\{\mathbf{s}_i, \mathbf{y}_i\}_{i=k-m}^{k-1}$ requires $\mathcal{O}(md)$ space complexity.
- Whether L-BFGS can also achieve super linear convergence is still unclear.
- The idea also works for SR1 and DFP.

Summary

method	convergence	time complexity	space complexity
Newton's method	quadratic	$O(d^3)$	$O(d^2)$
SR1/DFP/BFGS	super linear	$O(d^2)$	$O(d^2)$
L-BFGS	linear or super linear?	$O(md)$	$O(md)$
GD/AGD	linear	$O(d)$	$O(d)$

Table: Convergence property for strongly convex functions