Comp6211e: Optimization for Machine Learning

Tong Zhang

Lecture 9: General Unconstrained Convex Optimization

Convex Optimization

In this lecture, we consider the general unconstrained convex optimization problem:

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

To characterize f, we consider

- Strong convexity: parameter λ (lower bound of Hessian)
- Smoothness: parameter L (upper bound of Hessian)
- Lipschitz (non-smooth): parameter G (upper bound of gradient)

Non-Smooth and Strongly Convex Problem

Assume that $f(x)$ is non-smooth but G-Lipschitz, and λ strongly convex. What is the convergence rate?

Example

We consider the SVM formulation

$$\min_w f(w) := \left[\frac{1}{n}\sum_{i=1}^{n}(1 - w^\top x_i y_i)_+ + \frac{\lambda}{2}\|w\|_2^2\right]$$

Subgradient Method

Algorithm 1: Subgradient Descent Method

Input: $f(x)$, $x_0, \eta_1, \eta_2, \ldots$
Output: x_T
for $t = 1, \ldots, T$ **do**
    Let $x_t = x_{t-1} - \eta_t g_t$, where $g_t \in \partial f(x_{t-1})$ is a subgradient
Return: x_T

Convergence for Strongly Convex and Nonsmooth Optimization

Theorem

Assume $f(x)$ is λ-strongly convex, and G-Lipschitz. Let $\eta_t = 1/(\lambda t)$, then we have

$$\frac{1}{T}\sum_{t=1}^{T} f(x_{t-1}) \leq \min_x f(x) + \frac{(\ln T + 1)G^2}{2\lambda T}.$$

Proof

Given any x, we have

$$\begin{aligned}
\|x_t - x\|_2^2 =& \|(x_t - x_{t-1}) + (x_{t-1} - x)\|_2^2 \\
=& \|x_t - x_{t-1}\|_2^2 + 2(x_t - x_{t-1})^\top (x_{t-1} - x) + \|x_{t-1} - x\|_2^2 \\
=& \eta_t^2 \|g_t\|_2^2 - 2\eta_t g_t^\top (x_{t-1} - x) + \|x_{t-1} - x\|_2^2 \\
\leq& \|x_{t-1} - x\|_2^2 + 2\eta_t g_t^\top (x - x_{t-1}) + \eta_t^2 G^2 \\
\leq& \|x_{t-1} - x\|_2^2 + 2\eta_t \left[f(x) - f(x_{t-1}) - \frac{\lambda}{2}\|x - x_{t-1}\|_2^2 \right] + \eta_t^2 G^2.
\end{aligned}$$

Proof (continue)

Dividing by η_t^{-1}, we obtain

$$\frac{1}{\eta_t}\|x_t - x\|_2^2 \leq \left(\frac{1}{\eta_t} - \lambda\right)\|x_{t-1} - x\|_2^2 + 2[f(x) - f(x_{t-1})] + \eta_t G^2.$$

By summing over $t = 1$ to T, we obtain

$$\lambda T\|x_T - x\|_2^2 \leq 2\sum_{t=1}^{T}[f(x) - f(x_{t-1})] + \sum_{t=1}^{T}\frac{G^2}{\lambda t}.$$

Smoothing

For a non-smooth but G Lipschitz function, we may smooth it and obtain an (L, ϵ)-smooth approximation that is $L = G^2/2\epsilon$ smooth.

The condition number of the smoothed objective is L/λ.

By applying the strong convex version of Nesterov's acceleration algorithm, we obtain convergence to ϵ-accuracy in

$$T = O(\sqrt{L/\lambda}\log(1/\epsilon)) = O(G/\sqrt{\lambda\epsilon}\log(1/\epsilon))$$

number of iterations.

Reduction to Smooth and Strongly Convex Solver

Assume that we have an optimization algorithm $\mathcal{A}$ for L-smooth and λ-strongly convex optimization, then we can use it to solve optimization for the other three situations.
We specifically consider

- $\mathcal{A}$ as gradient descent method
- $\mathcal{A}$ as accelerated gradient descent method

Smooth and Non-Strongly Convex Problem

Assume $f(x)$ is L-smooth but not strongly convex . Then given an accuracy ϵ, we may use solver $\mathcal{A}$ to solve the following problem

$$\min_x \tilde{x}(x), \qquad \tilde{f}(x) = f(x) + \frac{\epsilon}{2}\|x - x_0\|_2^2.$$

This function is $L + \epsilon$-smooth and $\lambda = \epsilon$ strongly convex.
If we use gradient descent, then in order to achieve ϵ accuracy, we need

$$\tilde{O}(L/\epsilon)$$

iterations.
If we use gradient descent, then in order to achieve ϵ accuracy, we need

$$\tilde{O}(\sqrt{L/\epsilon}).$$

Non-Smooth and Strongly Convex Problem

If $f(x)$ is non-smooth but G-Lipschitz, and λ-strongly convex, then one can smooth f, to obtain $\tilde{f}(x)$ that is (L, ϵ)-smooth approximation, and at least $\lambda/(1 + \lambda/L)$ strongly convex.

This leads to a smoothed objective function with condition number of $O(G^2/(\epsilon\lambda)$.
We can apply a smooth and strongly convex solver.
We set learning rate as $O(1/L)$ and set $\beta = (1 - \sqrt{\alpha\lambda})/(1 + \sqrt{\alpha\lambda})$ or set adaptively.

Non-Smooth and Non-Strongly Convex Problem

We can find $\tilde{f}$ such that

$$\tilde{f}(x) = \min_z \left[f(z) + \frac{L}{2}\|x - z\|_2^2 \right] + \frac{\epsilon}{2}\|x - x_0\|_2^2.$$

This gives $L = (G^2/2\epsilon) + \epsilon$-smooth and ϵ-strongly convex.

This gives a condition number of $O(G^2/\epsilon^2)$.
We can apply a smooth and strongly convex solver.

Summary: Gradient Descent

	smooth	nonsmooth
strongly-convex	$\tilde{O}(L/\lambda)$ gradient descent	$\tilde{O}(G^2/\lambda\epsilon)$ sub-gradient
non-strongly-convex	$\tilde{O}(L/\epsilon)$ gradient descent	$\tilde{O}(G^2/\epsilon^2)$ sub-gradient

Table: Optimization Complexity for Gradient Descent

Accelerated Gradient Descent

	smooth	nonsmooth
strongly-convex	$\tilde{O}(\sqrt{L/\lambda})$ accelerated gradient	$\tilde{O}(G/\sqrt{\lambda\epsilon})$ accelerated gradient with smoothing
non-strongly-convex	$\tilde{O}(\sqrt{L/\epsilon})$ accelerated gradient	$\tilde{O}(G/\epsilon)$ accelerated gradient with smoothing

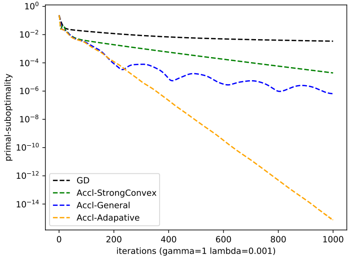
Table: Optimization Complexity for Accelerated Gradient Descent

Empirical Study

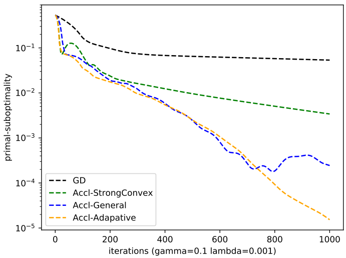
We study the effect of smoothing for gradient descent and accelerated gradient methods for SVM. We use a smoothing of the hinge loss for SVM, where the hinge loss $(1 - z)_+$ is replaced by

$$\phi_\gamma(z) = \max_z \left[(1 - z)_+ + \frac{1}{2\gamma}(x - z)^2 \right].$$

Empirical Results

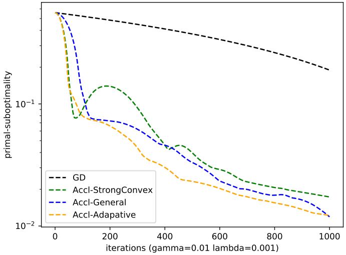


(a) $\gamma = 1$

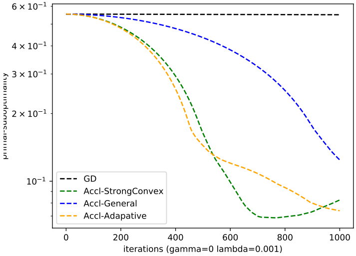


(b) $\gamma = 0.1$

Empirical Results



(a) $\gamma = 0.01$



(b) $\gamma = 0$

Summary

There are four cases categorized by strong-convexity and smoothness.

- Turn non-strongly convex into strongly convex function: add $\lambda = O(\epsilon)$ strongly convex regularizer.
- Turn non-smooth into smooth function function: $L = O(1/\epsilon)$ smooth.

Can apply solver for strongly convex and smooth functions. Can always set learning rate as $O(1/L)$.