Comp6211e: Optimization for Machine Learning

Tong Zhang

Lecture 7: Adaptive and General Acceleration Methods

Convex Optimization

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

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

Here we assume that $f(x)$ is an L-smooth convex function, and λ-strongly convex.

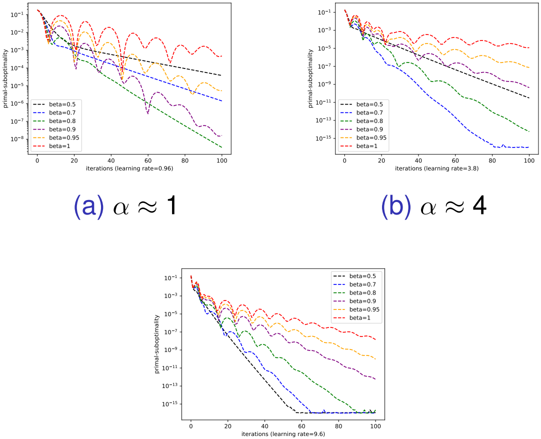
Nesterov's Acceleration

In the previous lectures, we have shown that for strongly convex problems, it is possible to improve the gradient descent method by using Nesterov's acceleration method, which is the following algorithm:

$$
\begin{aligned}
y_t &= x_{t-1} + \beta(x_{t-1} - x_{t-2}) \\
x_t &= y_t - \alpha \nabla f(y_t),
\end{aligned}
$$

with appropriately chosen α and β.

Sensitivity of β



(a) $\alpha \approx 1$

(b) $\alpha \approx 4$

(c) $\alpha \approx 10$

Figure: Convergence Comparisons with Fixed α

Adaptive Tuning of β

Recall the convergence theory.

Theorem

Assume $f(x)$ is L-smooth and λ-strongly convex. Let $\eta \leq 1/L$ and $\theta = \sqrt{\eta\lambda}$. Let $\alpha_t = \eta \leq 1/L$ and $\beta_t = \beta = (1-\theta)/(1+\theta)$. Then

$$f(x_t) \leq f(x_*) + (1-\theta)^t \left[f(x_0) - f(x_*) + \frac{\lambda}{2}\|z - x_0\|_2^2 \right]$$

.

Idea for Adaptation

- Start with large θ, set $\beta = (1 - \theta)/(1 + \theta)$
- Check convergence rate of the theorem
- If not satisfied, reduce θ

Practical Convergence Rate Checking

In practice, one of the most frequently used method is to check the convergence in terms of the gradient

$$\|\nabla f(x)\|_2$$

at a point x.
Therefore, we have to derive a convergence of gradients from convergence of function values.

Convergence Checking with Gradients

From Lecture 04, we have:

$$\frac{1}{2L}\|\nabla f(x)\|_2^2 \leq [f(x) - f(x_*)] \leq \frac{1}{2\lambda}\|\nabla f(x)\|_2^2.$$

We can obtain the following for the Nesterov's method: if $\theta \leq \sqrt{\eta\lambda}$, then we have

$$\|\nabla f(x_t)\|_2^2 \leq \frac{2}{\eta\lambda}(1-\theta)^t\|\nabla f(x_0)\|_2^2 \leq \frac{2}{\theta^2}(1-\theta)^t\|\nabla f(x_0)\|_2^2. \qquad (1)$$

This convergence rate can be checked.

Algorithm

Algorithm 1: Adaptive Acceleration Method (Theoretically Motivated)

```
Input: $f(x)$, $x_0$, $\alpha = \eta \leq 1/L$
Output: $x_T$
Let $x_{-1} = x_0$
Let $s = 0$
Let $\theta = 0.5$
Let $\beta = (1 - \theta)/(1 + \theta)$
Let $\Delta = \lceil \log(0.5\theta^2)/\log(1 - \theta) \rceil$
for $t = 1, \ldots, T$ do
    Let $y_t = x_{t-1} + \beta(x_{t-1} - x_{t-2})$
    Let $x_t = y_t - \alpha \nabla f(y_t)$
    if $(t - s)\%\Delta == 0$ and $\|\nabla f(x_t)\|_2^2 > 2\theta^{-2}(1 - \theta)^{t-s}\|\nabla f(x_s)\|_2^2$ then
        if $\|\nabla f(x_t)\|_2^2 \geq \|\nabla f(x_{t-\Delta})\|_2^2$ then
            Let $x_{t-1} = x_t = x_{t-\Delta}$
        else
            Let $x_{t-1} = x_t$
        Let $s = t$
        Let $\theta = \theta/2$
        Let $\beta = (1 - \theta)/(1 + \theta)$
        Let $\Delta = \lceil \log(0.5\theta^2)/\log(1 - \theta) \rceil$
Return: $x_T$
```

Convergence

Theorem

Assume that $f(x)$ is λ-strongly convex with $\lambda > 0$. Let $m = \lceil \log_2(0.5/\sqrt{\eta\lambda}) \rceil$, and let

$$T_0 = \left\lceil \frac{4(m+1)}{\sqrt{\eta\lambda}} \right\rceil,$$

then when $T \geq T_0$, we have

$$\|\nabla f(x_t)\|_2^2 \leq 2\bar{\theta}^{-2}(1-\bar{\theta})^{T-T_0}\|\nabla f(x_0)\|_2^2,$$

where $\bar{\theta} = 2^{-m-1} \geq 0.5\sqrt{\eta\lambda}$.

Practical Algorithm

The previous algorithm is too conservative in practice. We may choose a simpler and more aggressive method where we tune β using observed convergence rate, measured by

$$\gamma = \log \frac{\|\nabla f(y_t)\|_2^2}{\|\nabla f(y_{t-1})\|_2^2}.$$

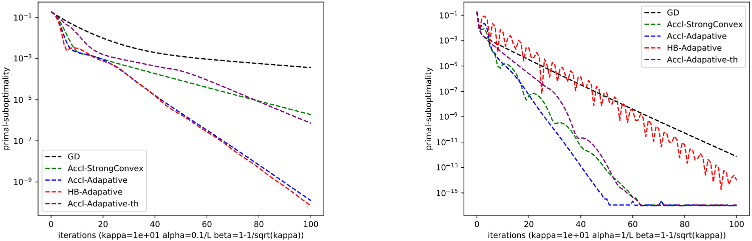
We can use the observed learning rate to set θ and γ accordingly. This performs better than Algorithm 1.

Adaptive Nesterov's Method (Practical)

Algorithm 2: Adaptive Acceleration Method (Practically Simplified)

```
Input: $f(x)$, $x_0$, $\alpha = \eta \leq 1/L$
Output: $x_T$
Let $x_{-1} = x_0$
Let $\gamma = 0$
Let $y_0 = x_0$
for $t = 1, \ldots, T$ do
    Let $\beta = \min(1, \exp(\gamma))$
    Let $y_t = x_{t-1} + \beta(x_{t-1} - x_{t-2})$
    Let $x_t = y_t - \alpha \nabla f(y_t)$
    Let $\gamma = 0.8\gamma + 0.2 \ln(\|\nabla f(y_t)\|_2^2 / \|\nabla f(y_{t-1})\|_2^2)$
Return: $x_T$
```

Empirical Study



(a) $\alpha = 0.1/L$

(b) $\alpha \approx 1/L$

Figure: Convergence Comparisons with Fixed Learning Rate

General Nesterov's Method

If the function is not strongly convex, then theory of Algorithm 1 does not apply.

However, one can modify the acceleration algorithm so that a theory can still be obtained even when $\lambda = 0$.

The general situation can be stated as an algorithm, which is applicable for the smooth and non-strongly convex case.

General Acceleration

```
Algorithm 3: Nesterov's Acceleration Method (Non Strongly Convex)
Input: $f(x)$, $x_0$, $\eta \leq 1/L$
       $\lambda \in [0, 1/\eta]$ (default is $\lambda = 0$)
       $\gamma_0 \in [\lambda, 1/\eta]$ (default is $\gamma_0 = 1/\eta$)
Output: $x_T$
Let $x_{-1} = x_0$
Let $\theta_0 = 1$
for $t = 1, \ldots, T$ do
    Solve for $\theta_t$: $\theta_t^2/\eta = \theta_t\lambda + (1 - \theta_t)\gamma_{t-1}$
    Let $\gamma_t = (1 - \theta_t)\gamma_{t-1} + \theta_t\lambda$
    Let $\beta_t = (\theta_t^{-1} - 1)(\theta_{t-1}^{-1} - 1)\gamma_{t-1}/(\eta^{-1} - \lambda)$
    Let $y_t = x_{t-1} + \beta_t(x_{t-1} - x_{t-2})$
    Let $x_t = y_t - \eta\nabla f(y_t)$
Return: $x_T$
```

Convergence

We will have the following general theorem.

Theorem

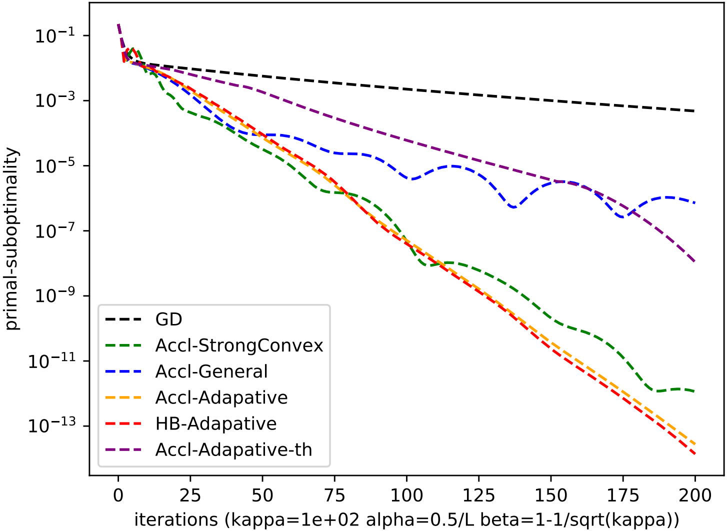
Assume $f(x)$ is L-smooth and λ-strongly convex. Then for all $x_* \in \mathbb{R}^d$, we have

$$f(x_t) \leq f(x_*) + \lambda_t \left[f(x_0) - f(x_*) + \frac{\gamma_0}{2} \|x_* - x_0\|_2^2 \right],$$

where

$$\lambda_t = \prod_{s=1}^{t} (1 - \theta_s).$$

Empirical Study



Summary

We have studied the momentum term β for the accelerated first order methods

- We should adjust β from small value to large value (or θ from large value to small value)

In theory, we can tune β with theoretical guarantees

- β can be set adaptively from large to small (adaptive method)
- β can be set manually from large to small (general method)

In practice, there is simple method to tune β adaptively (but no formal theoretical guarantees)

- β can be set according to observed learning rate