Modern Computational Statistics lec03 Notes
Part I

This is based on PKU course: Modern Computational Statistics. Thanks to Prof. Cheng Zhang, this is a very interesting course.

This lecture will introduce some advanced gradient descent methods(which is some kind of hard). In Part I, I will introduce the upper bound convergence rate of gradient descent and help you understand the Nestrov's accelaration method.

1 Some Problems with Traditional Gradient Descent

Even though traditional gradient descent is used in many algorithms, it still has some flaws.

1.1 Saddle Point Problem

It's easy for us to use gradient descent while solving convex functions, because they usually have just one local minimum. However, for non-convex problems, it's more complicated.

1.2 Not applicable to non-differential objectives

It's very obvious that for a non-differential objective, it's almost impossible to take the derivatives.

1.3 Could be slow

Even if we used to say that gradient descent can converges, but it is still a NP-Hard problem to solve the objective in polynomial time.

2 Momentum Method

Momentum method is introduced by Polyak in 1964. It can accelerate gradient descent by taking accounts of previous gradients in the update rule at each iteration.

Recall the original update function of gradient descent:

$$\begin{aligned} W &:= W - \alpha \nabla W \\ b &:= b - \alpha \nabla b \end{aligned} \tag{1}$$

The gradient is only about the current gradient, but not the previous ones. Momemtum Method deals with this problem by using previous parameters.

For example, in iteration 100, we get the series of the gradients:

$$\{\nabla W_1, \nabla W_2, ..., \nabla W_{100}\}$$

Thus, the momentum gradients are:

$$\begin{aligned} V_{\nabla W_0} &= 0 \\ V_{\nabla W_1} &= \beta V_{\nabla W_0} + (1-\beta)\nabla W_1 \\ V_{\nabla W_2} &= \beta V_{\nabla W_1} + (1-\beta)\nabla W_2 \\ &... \\ V_{\nabla W_{100}} &= \beta V_{\nabla W_{99}} + (1-\beta)\nabla W_{100} \end{aligned} \tag{2}$$

where $\beta \leq \mu < 1$.

See Andrew's course for more details in exponential weight averages and mini-batch gradient descent.

3 Nesterov's Acceleration

Choose any initial $x^{(0)} = x^{(-1)}, \forall k = 1, 2, 3...$,

$$\begin{aligned} y &= x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-)1} - x^{(k-2)}) \\ x^{(k)} &= y - t_k \nabla f(y) \end{aligned} \tag{3}$$

It's always said that nesterov's acceleration is some kind of mysterious, because the simple transformation does work. The first two steps are the usually gradient updates. After that, $y = x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-)1} - x^{(k-2)}$ carries some momentum from previous iterations, and $x^{(k)} = y - t_k \nabla f(y)$ uses lookahead gradient at y.

4 Convergence Rate of Gradient Methods

4.1 What is convergence rate?

For a optimization problem of first order, we always want to get the convergence rate of the following three conditions:

- How many times do we have to iterate to get the x_T close enough to x^*? i.e.

$$||x_T - xx^*|| \leq \epsilon$$

- How many times do we have to iterate to get the derivative $\nabla f(x_T)$ less than ϵ? i.e.

$$||\nabla f(x_T)|| \leq \epsilon$$

- How many times do we have to iterate to get the value $f(x_T)$ close enough to f^*

For example, if we want $||x_T - x^*|| \leq \epsilon$, and the iteration time is T, and $T \geq \frac{1}{\epsilon}$, than we say that the iteration times should be at least $O(\frac{1}{\epsilon})$. ϵ is always considered as a very small number, so $\frac{1}{\epsilon} \leq \frac{1}{\epsilon^2}$, meaning that the previous algo runs always faster.

4.1.1 Sublinear Convergence

We can always get a recursive formula like

$$||x_T - x^*|| \leq \frac{1}{T^{1/k}} ||x_0 - x^*||$$

Let $\frac{1}{T^{1/k}} ||x_0 - x^*|| \leq \epsilon$(suppose $C = ||x_0 - x^*||$), then $T \geq \frac{C}{\epsilon^k}$.

So, sublinear convergence is of $O(\frac{1}{\epsilon^k})$.

4.1.2 Linear Convergence

We can always get a recursive formula like

$$||x_{t+1} - x^*|| \leq q||x_t - x^*||$$

And after recursion:

$$||x_T - x^*|| \leq q^T ||x_0 - x^*||$$

Similarly, take the log of $q^T ||x_0 - x^*|| \leq \epsilon$, we get:

$$T \log q + \log ||x_0 - x^*|| \leq \log(epsilon)$$

Divided by $\log q$,

$$T \geq C \log(\frac{1}{\epsilon})$$

So, linear convergence is of $O(\log(\frac{1}{\epsilon}))$.

4.1.3 Quadratic Convergence

We can always get a recursive formula like

$$||x_{t+1} - x^*|| \leq q||x_t - x^*||^2$$

After recursion:

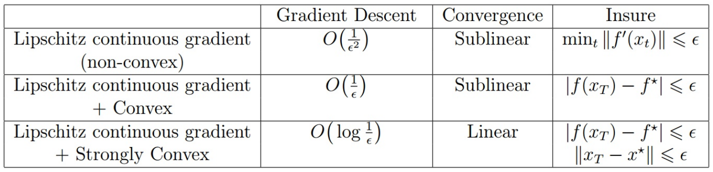
$$\begin{aligned}
||x_T - x^*|| &\leq q||x_{T-1} - x^*||^2 \\
&\leq qq^2||x_{T-2} - x^*||^4 \\
&\leq qq^2...q^{2(T-1)}||x_0 - x^*||^{2^T} \\
&\leq q^{\frac{1-2^T}{1-2}}||x_0 - x^*||^{2^T} \\
&\leq q^{2^T-1}||x_0 - x^*||^{2^T} \\
&\leq Cq^{2^T}||x_0 - x^*||^{2^T} (q \quad constant) \\
&\leq C(q||x_0 - x^*||)^{2^T}
\end{aligned} \tag{4}$$

Take the log, we finally get:

$$T \geq C \log\log\frac{1}{\epsilon}$$

So, quadratic convergence is of $O(\log\log(\frac{1}{\epsilon}))$.

4.1.4 The convergence rate of gradient descent



	Gradient Descent	Convergence	Insure
Lipschitz continuous gradient (non-convex)	$O(\frac{1}{\epsilon^2})$	Sublinear	$\min_t \|f'(x_t)\| \leqslant \epsilon$
Lipschitz continuous gradient + Convex	$O(\frac{1}{\epsilon})$	Sublinear	$\|f(x_T) - f^\star\| \leqslant \epsilon$
Lipschitz continuous gradient + Strongly Convex	$O(\log\frac{1}{\epsilon})$	Linear	$\|f(x_T) - f^\star\| \leqslant \epsilon$ $\|x_T - x^\star\| \leqslant \epsilon$

Table 2: GD Convergence Under Different Assumption

Figure 1:

This table will help understand the convergence rate of gradient descent under different conditions.

4.2 Introduction to the convergence rate of gradient descent

Some Assumptions:

- f is convex and continuously differentiable on R^n
- $\nabla f(x)$ is L-Lipschitz continuous w.r.t. Euclidean norm: for any $x, y \in R^n$

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||$$

- $x^* = argmin \quad f(x), f^* = \inf_x f(x)$

Theorem 1 *Gradient descent with $0 < t \leq 1/L$ satisfies:*

$$f(x^{(k)}) - f^* \leq \frac{1}{2kt}||x^{(0)} - x^*||^2$$

4.3 Some Useful Lemma and Strong Convexity

Let's recall some L-lipschitz continuous property:
If $\nabla f(x)$ is L-lipschitz continuous, then:

- $||\nabla f(x) - \nabla f(y)||_2 \leq L||x - y||_2$
- $\frac{L}{2}x^T x - f(x)$ is convex
- $\nabla^2 f(x) \leq LI$
- $f(y) \leq f(x) + \nabla f(x)(y - x) + \frac{L}{2}||y - x||_2^2$ if f(x) is convex. We will use this property later!

And also, if f is differentiable and m-strongly convex, then

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{m}{2}||y - x||^2$$

4.4 Proof of Theorem1

Let $x^+ = x - t\nabla f(x)$and $0 < t \leq 1/L$, then according to L-lipschitz's continuous(take x^+ as y), then we get:

$$f(x^+) \leq f(x) - t||\nabla f(x)||^2 + \frac{t^2 L}{2}||\nabla f(x)||^2$$

$$\leq f(x) - \frac{t}{2}||\nabla f(x)||^2$$

And because of convexity(see the above equation),

$$f(x) \leq f^* + \nabla f(x)^T(x - x^*)\frac{m}{2}||x - x^*||^2$$

Adding the above two inequalities

$$\begin{aligned} f(x^+) - f^* &\leq \nabla f(x)^T(x - x^*) - \frac{t}{2}||\nabla f(x)||^2 - \frac{m}{2}||x - x^*||^2 \\ &\leq \frac{1}{2t}(||x - x^*||^2 - ||x^+ - x^*||^2)\frac{m}{2}||x - x^*||^2 \\ &= \frac{1}{2t}((1 - mt)||x - x^*||^2 - ||x^+ - x^*||^2) Trytodecompositethepreviousequation \\ &\leq \frac{1}{2t}(||x - x^*||^2 - ||x^+ - x^*||^2) \end{aligned} \tag{5}$$

For gradient descent updates:

$$\sum_{i=1}^{k}(f(x^{(i)}) - f^*) \leq \frac{1}{2t}\sum_{i=1}^{k}(||x^{(i-1)} - x^*||^2 - ||x^{(i)} - x^*||^2)$$

$$= \frac{1}{2t}(||x^{(0)} - x^*||^2 - ||x^{(k)} - x^*||^2)$$

Since $f(x^i)$ is non-increasing:

$$f(x^{(k)}) - f^* \leq \frac{1}{2kt}||x^{(0)} - x^*||^2$$

If f is $m - strongly$ convex, and $m > 0$,

$$||x^{(i)} - x^*||^2 \leq (1 - mt)||x^{(i-1)} - x^*||^2, \forall i = 1, 2, ...$$

(however I really cant understand where this comes from) Therefore,

$$||x^{(k)} - x^*||^2 \leq (1 - mt)^k||x^{(0)} - x^*||^2$$

Here let's write the theorem again:

Theorem 2 *Gradient descent with $0 < t \leq 1/L$ satisfies:*

$$f(x^{(k)}) - f^* \leq \frac{1}{2kt}||x^{(0)} - x^*||^2$$

4.5 Oracle Lower Bound of First-order Methods

First order methods refer to methods with first gradient, i.e., any iterative algorithm that selects x^{k+1} in the set

$$x^{(0)} + span\{\nabla f(x^{(0)}), \nabla f(x^{(1)}), ..., \nabla f(x^{(k)})\}$$

span: space that composed by vector's linear combinations

Theorem 3 *(Nesterov:) For every integer $k \leq (n-1)/2$ and every $x^{(0)}$, there exists functions that satisfy the assumptions such that for any first-order method:*

$$f(x^{(k)}) - f^* \geq \frac{3}{32}\frac{L||x_0 - x^*||^2}{(k+1)^2}$$

Therefore, this proves that $1/k^2$ is the best convergence rate for all first-order methods.

And, we can calculate the convergence rate of Nesterov's Acceleration:

$$f(x^{(k)}) - f^* \leq \frac{2||x^{(0)} - x^*||^2}{t(k+1)^2}$$

We can see that, they are all of $1/k^2$, so nesterov's acceleration always achieves the best convergence rate of gradient descent.