1 Introduction

The gradient descent algorithm is one of the most popular optimization algorithms. This note introduces three types of gradient descent algorithms: (standard) gradient descent, projected gradient descent, and proximal gradient descent.

We are interested in the following optimization problem

\[
\min_x \quad f(x) + g(x)
\]

s.t. \( x \in \Omega \),

where \( f(x) \) is a smooth convex function, \( g(x) \) is a closed convex function, and \( \Omega \) is a closed convex set. The following introduces some basic definitions in optimization.

The key idea of many optimization methods can be explained as constructing a model function to approximate the smooth function \( f(x) \) at every iterate \( x_k \). The model function defined for the gradient descent method at iterate \( x_k \) is

\[
m_{x_k, \gamma_k}(x) := f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\gamma_k} \| x - x_k \|^2.
\]

There are multiple reasons why \( m_{x_k, \gamma_k} \) is a good approximation of \( f(x) \):

- \( m_{x_k, \gamma_k}(x_k) = f(x_k) \);
- \( m'_{x_k, \gamma_k}(x_k) = \nabla f(x_k) \);
- If \( \gamma_k > 0 \) is small enough, then \( f(x) \leq m_{x_k, \gamma_k}(x) \).

The model function is usually easy to minimize even associating with a constraint or an additional function \( g(x) \).

2 Gradient Descent

We start from the unconstrained smooth convex optimization:

\[
\min_x \quad f(x)
\]
where $f(x)$ is convex smooth. The gradient descent algorithm updates the variable $x$ by minimizing the model function at each iteration:

$$x_{k+1} = \arg\min_x m_{x_k, \gamma_k}(x) = \arg\min_x \frac{1}{2 \gamma_k} \|x - x_k + \gamma_k \nabla f(x_k)\|^2 = x_k - \gamma_k \nabla f(x_k).$$

Since $x_{k+1}$ minimize $m_{x_k, \gamma_k}(x)$, we have the following important property (if $\gamma_k$ is small enough):

$$f(x_{k+1}) \leq \min_x m_{x_k, \gamma_k}(x) \leq m_{x_k, \gamma_k}(x_k) = f(x_k). \quad (1)$$

The gradient algorithm is summarized in Alg. 1.

**Algorithm 1 Gradient Descent**

Require: $x_0, \{\gamma_k\}, K$
Ensure: $x_K$

$k = 0$
while $k \neq 0$ do
    $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$
    $k \leftarrow k + 1$
end while

One safe way to choose $\gamma_k$ to ensure (1) is letting

$$\gamma_k \leq \frac{1}{\max_x \|\nabla^2 f(x)\|}.$$

Another way is using line search: start from a large steplength $\gamma$ and decrease $\gamma$ if $f(x_{k+1}) \leq m_{x_k, \gamma_k}(x_{k+1})$ is not satisfied. The complete algorithm is summarized in Alg. 2.

**Algorithm 2 Gradient Descent (line search)**

Require: $x_0, K, \gamma, \beta \in (0, 1)$
Ensure: $x_K$

$k = 0$;
Initialize the step length $\gamma$;
while $k \neq 0$ do
    while TRUE do
        $x_{k+1} = \arg\min_x m_{x_k, \gamma_k}(x) = x_k - \gamma \nabla f(x_k)$
        if $f(x_{k+1}) \leq m_{x_k, \gamma_k}(x_{k+1})$ then break;
    end if
    $\gamma = \gamma \beta$;
end while
$k \leftarrow k + 1$
end while
3 Projected Gradient Descent

Now we consider the constrained smooth convex optimization:

$$\min_{x \in \Omega} f(x)$$

where $f(x)$ is convex smooth and $\Omega$ is a convex closed set. The projected gradient descent algorithm updates the variable $x$ by minimizing the model function at each iteration:

$$x_{k+1} = \arg\min_{x \in \Omega} m_{x_k, \gamma_k}(x) = \arg\min_{x \in \Omega} \frac{1}{2\gamma_k} \|x - x_k + \gamma_k \nabla f(x_k)\|^2$$

$$= P_{\Omega}(x_k - \gamma_k \nabla f(x_k)),$$

where $P_{\Omega}(\cdot)$ is defined by

$$P_{\Omega}(y) = \arg\min_{x \in \Omega} \|x - y\|^2.$$

One can easily plug the projected gradient step into Algs. 1 and 2.

4 Proximal Gradient Descent

Now we consider the constrained smooth convex optimization:

$$\min_x f(x) + g(x)$$

where $f(x)$ is convex smooth and $g(x)$ is a convex closed function. The proximal gradient descent algorithm updates the variable $x$ by minimizing the model function at each iteration:

$$x_{k+1} = \arg\min_{x} m_{x_k, \gamma_k}(x) + g(x) = \arg\min_{x} \frac{1}{2\gamma_k} \|x - x_k + \gamma_k \nabla f(x_k)\|^2 + g(x)$$

$$= \arg\min_{x} \frac{1}{2} \|x - x_k + \gamma_k \nabla f(x_k)\|^2 + \gamma_k g(x)$$

$$= \text{Prox}_{\gamma_k g(\cdot)}(x_k - \gamma_k \nabla f(x_k)),$$

where $\text{Prox}_{g(\cdot)}(\cdot)$ is defined by

$$\text{Prox}_{g(\cdot)}(y) = \arg\min_x \frac{1}{2} \|x - y\|^2 + g(x).$$

When $g(x)$ has a simple form such as $\|x\|$ and $\|x\|_1$ has a closed form:

$$\text{Prox}_{\|\cdot\|}(y) = \max \left(0, 1 - \frac{\lambda}{\|y\|}\right) y$$

$$\text{Prox}_{\|\cdot\|_1}(y) = \text{sgn}(y) \odot \max(0, |y| - \lambda),$$
where \( \text{sgn}(y) \) returns a sign vector of \( y \) (positive elements correspond to 1, negative elements correspond to \(-1\), and 0 elements correspond to 0), \( \odot \) defines a component-wise multiplication, and \( |y| \) returns a vector with absolute values of all elements in \( y \).

One may noticed that if \( g(x) \) is defined as an indicator function on a convex closed set \( \Omega \), that is,

\[
g(x) := \begin{cases} 0 & \text{if } x \in \Omega; \\ \infty & \text{otherwise} \end{cases}
\]

the proximal operation in (3) is nothing but the projection operator defined in (2).

To decide the steplength, one can safely choose \( \gamma_k \) as a positive value small enough, that is, \( \gamma_k \leq (\|\nabla^2 f(x)\|^{-1}) \forall x \). The other option is to use the line search scheme used in the gradient descent algorithm. However, due to the nonsmooth part in the objective function, we apply the line search search scheme used before. See Alg. 3 for the line search proximal gradient descent algorithm. Usually the line search scheme requires less iterations to converge.

---

**Algorithm 3 Proximal Gradient Descent (line search)**

**Require:** \( x_0, K, \gamma, \beta \in (0, 1) \)

**Ensure:** \( x_K \)

\( k = 0; \)

Initialize the step length \( \gamma; \)

while \( k \neq 0 \) do

while TRUE do

\[
x_{k+1} = \arg \min_x m_{x_k, \gamma}(x) + g(x) = \text{prox}_{\gamma g(\cdot)}(x_k - \gamma \nabla f(x_k))
\]

if \( f(x_{k+1}) \leq m_{x_k, \gamma}(x_{k+1}) \) then

break;

end if

\( \gamma = \gamma \beta; \)

end while

\( k \leftarrow k + 1 \)

end while

---

5 Convergence Rate

It is easy to see three methods can monotonically decrease the objective function \( f(x) \), which suggests the convergence. However, the convergence rate is unclear. This section introduces the convergence rates of three algorithms. The other option is to apply the line search scheme (used in gradient descent algorithm) to adaptively decide the steplength. Usually the line search scheme requires less iterations to converge, see

**Assumption 1.** Assume that \( f(x) \) is a convex smooth function and has a Lipschitz gradient, that is, there exists a constant \( L \) such that

\[
f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2 \quad \forall x, y.
\]
Assumption 2. Assume that \( f(x) \) is a strongly smooth convex function, that is, there exists a constant \( L \) such that
\[
f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \| y - x \|^2 \quad \forall x, y.
\]
If Assumption 1 is satisfied, we have sublinear convergence rate:
\[
f(x_k) - f^* \leq O\left(\frac{1}{k}\right).
\]
If both Assumptions 1 and 2 are satisfied, we have linear convergence rate:
\[
f(x_k) - f^* \leq O\left((1 - l/L)^k\right).
\]

Proof. From Assumption 1, we have
\[
f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \| x_{k+1} - x_k \|^2
\]
\[
= f(x_k) - \frac{1}{L} \| \nabla f(x_k) \|^2 + \frac{L}{2} \| x_{k+1} - x_k \|^2
\]
\[
= f(x_k) - \frac{1}{2L} \| \nabla f(x_k) \|^2.
\]
Let \( \Delta_k = f(x_k) - f^* \). Next we prove \( \Delta_k \geq \frac{l}{2} \| x_k - x^* \|^2 \). For any \( x \), we have
\[
f(x) - f(x_k) \geq \langle \nabla f(x_k), x - x_k \rangle + \frac{l}{2} \| x - x_k \|^2.
\]
Therefore, we have
\[
\min_x f(x) - f(x) \geq \min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{l}{2} \| x - x_k \|^2 = \frac{1}{2l} \| \nabla f(x_k) \|^2.
\]
It follows that
\[
\Delta_{k+1} \leq \Delta_k - \frac{1}{2L} \| \nabla f(x_k) \|^2 \leq \Delta_k - \frac{l}{L} \Delta_k = (1 - l/L)\Delta_k.
\]
It proves the linear rate for strongly convex functions.

Next we prove for the weakly convex functions. We have from the convexity
\[
\Delta_k \leq \langle \nabla f(x_k), x_k - x^* \rangle \leq \| x_k - x^* \| \| \nabla f(x_k) \| \leq \| x_0 - x^* \| \| \nabla f(x_k) \|.
\]
The last inequality is due to the monotonicity of \( \| x_k - x^* \| \):
\[
\| x_k - x^* \|^2 = \| x_k - \frac{1}{L} \nabla f(x_k) - x^* \|^2
\]
\[
= \| x_k - x^* \|^2 + \frac{1}{L^2} \| \nabla f(x_k) \|^2 - 2 \frac{L}{L} \langle \nabla f(x_k), x_k - x^* \rangle
\]
\[
\leq \| x_k - x^* \|^2 + \frac{1}{L^2} \| \nabla f(x_k) \|^2 - 2 \frac{L}{L^2} \| \nabla f(x_k) \|^2
\]
\[
= \| x_k - x^* \|^2 - \frac{1}{L^2} \| \nabla f(x_k) \|^2.
\]
Therefore, we have the following recursion:

\[ \Delta_{k+1} \leq \Delta_k - C\Delta_k^2 \]

where \( C = \frac{1}{2L\|x_0 - x^*\|} \). It follows

\[ \frac{1}{\Delta_k} \leq \frac{1}{\Delta_{k+1}} - C \frac{\Delta_k}{\Delta_{k+1}} \]

which suggests

\[ \frac{1}{\Delta_{k+1}} \geq \frac{1}{\Delta_k} + C \geq \cdots \geq \frac{1}{\Delta_0} + (k + 1)C. \]

It proves the sublinear rate \( \Delta_k \leq O(1/k). \)