Computational Methods in Optimization  
Spring 2016, Subgradient methods  
Jean Honorio jhonorio@purdue.edu  

1 Subgradient descent method

For brevity, everywhere differentiable functions will be called smooth. Similarly, not everywhere differentiable functions will be called nonsmooth.

First, we define Lipschitz continuity.

**Definition 1** (Lipschitz continuity). A function $\phi : \mathbb{R}^p \to \mathbb{R}$ is called $K$-Lipschitz continuous with respect to the norm $\| \cdot \|$ if and only if there is a constant $K < +\infty$ such that

$$(\forall x, u \in \mathbb{R}^p) \ |\phi(x) - \phi(u)| \leq K \|x - u\|$$

A smooth function $\phi : \mathbb{R}^p \to \mathbb{R}$ is called $K$-Lipschitz continuous with respect to the norm $\| \cdot \|$ if and only if there is a constant $K < +\infty$ such that

$$(\forall x \in \mathbb{R}^p) \ \|\nabla \phi(x)\| \leq K$$

Recall that the gradient of a smooth convex function $\phi : \mathbb{R}^p \to \mathbb{R}$ at $x$ fulfills:

$$(\forall u \in \mathbb{R}^p) \ \phi(u) - \phi(x) \geq \langle \nabla \phi(x), u - x \rangle$$

**Definition 2** (Subgradient). For a (possibly nonsmooth) convex function $\phi : \mathbb{R}^p \to \mathbb{R}$, we can define a subdifferential set as follows:

$$\partial \phi(x) = \{ g \mid (\forall u \in \mathbb{R}^p) \ \phi(u) - \phi(x) \geq \langle g, u - x \rangle \}$$

Each element $g \in \partial \phi(x)$ is called a subdifferential or subgradient of $\phi$ at $x$.

Clearly, in the above definition, if $\phi : \mathbb{R}^p \to \mathbb{R}$ is smooth, then $\partial \phi(x)$ has a single element for every $x \in \mathbb{R}^p$. If $\partial \phi(x)$ is nonsmooth there exist some $x \in \mathbb{R}^p$ for which $\partial \phi(x)$ has more than one element.

Consider for instance the nonsmooth function $\phi(w) = |w|$ where $w \in \mathbb{R}$. By Definition 2, we have:

$$\partial \phi(0) = \{ g \mid (\forall u \in \mathbb{R}) \ |u| \geq g \ u \}$$

Thus, clearly $\partial \phi(0) = [-1, +1]$. 

1
Now, consider the following optimization problem where \( f : \mathbb{R}^p \to \mathbb{R} \) is convex and \( K \)-Lipschitz with respect to the \( \ell_2 \)-norm:

\[
\hat{x} = \arg \min_{x \in \mathbb{R}^p} f(x) \quad (1)
\]

Let \( \eta_t \) be the step size at iteration \( t \geq 1 \). Specifically, let \( \beta \) be a constant factor and define:

\[
\eta_t = \frac{\beta}{K \sqrt{t}}
\]

Consider the next subgradient descent algorithm for solving the above problem:

**Algorithm 1** Subgradient descent algorithm

**Input:** Number of iterations \( T \geq 1 \), factor \( \beta > 0 \), initial point \( x^{(1)} \in \mathbb{R}^p \) (The setting of \( x^{(1)} \) can be uninformed, e.g., \( x^{(1)} = 0 \))

**for** \( t = 1 \ldots T - 1 \) **do**

\( x^{(t+1)} \leftarrow x^{(t)} - \eta_t g^{(t)} \) where \( g^{(t)} \in \partial f(x^{(t)}) \)

**end for**

**Output:** \( \tilde{x}^{(T)} \leftarrow \frac{\sum_{t=1}^{T} \eta_t x^{(t)}}{\sum_{t=1}^{T} \eta_t} \)

### 2 Convergence analysis

In what follows, we state our main result regarding convergence rates for Algorithm 1.

**Theorem 1** (Adapted from [1]). Assume that \( f : \mathbb{R}^p \to \mathbb{R} \) is convex and \( K \)-Lipschitz with respect to the \( \ell_2 \)-norm in the problem of eq. (1). Recall that \( \tilde{x} \) is the optimal solution of the problem of eq. (1). Assume that Algorithm 1 runs for a number of iterations \( T \), factor \( \beta \) and initial point \( x^{(1)} \), and that the algorithm outputs \( \tilde{x}^{(T)} \). We have:

\[
f(\tilde{x}^{(T)}) - f(\tilde{x}) \leq \frac{K\|x^{(1)} - \tilde{x}\|_2^2}{4\beta(\sqrt{T} - 1)} + \frac{\beta K(1 + \log T)}{4(\sqrt{T} - 1)}
\]

**Proof.** Let \( d^{(t)} = \|x^{(t)} - \tilde{x}\|_2^2 \). Note that since \( g^{(t)} \in \partial f(x^{(t)}) \), by Definition 2 we have:

\[
f(\tilde{x}) - f(x^{(t)}) \geq \langle g^{(t)}, \tilde{x} - x^{(t)} \rangle
\]
In fact, the above holds \( \forall \hat{x} \in \mathbb{R}^p \) but in our problem we care about a unique \( \hat{x} \).

By the Lipschitz continuity of \( f \), we know that (\( \forall t \)) \( \| g(t) \|_2 \leq K \). Therefore:

\[
a^{(t+1)} = \| x^{(t+1)} - \hat{x} \|_2^2 \\
= \| (x^{(t)} - \hat{x}) - \eta_t g^{(t)} \|_2^2 \\
= \| x^{(t)} - \hat{x} \|_2^2 - 2\eta_t (g^{(t)}, x^{(t)} - \hat{x}) + \eta_t^2 \| g^{(t)} \|_2^2 \\
\leq a^{(t)} + 2\eta_t \left( f(\hat{x}) - f(x^{(t)}) \right) + \eta_t^2 K^2
\]

Reorganizing the above, we obtain:

\[
\eta_t \left( f(x^{(t)}) - f(\hat{x}) \right) \leq \frac{1}{2} \left( a^{(t)} - a^{(t+1)} + \eta_t^2 K^2 \right)
\]

Summing over \( t \), we get:

\[
\sum_{t=1}^T \eta_t \left( f(x^{(t)}) - f(\hat{x}) \right) \leq \frac{1}{2} \sum_{t=1}^T \left( a^{(t)} - a^{(t+1)} + \eta_t^2 K^2 \right) \\
= \frac{1}{2} \left( a^{(1)} - a^{(T+1)} + K^2 \sum_{t=1}^T \eta_t^2 \right) \\
\leq \frac{1}{2} \left( \| x^{(1)} - \hat{x} \|_2^2 + \beta^2 \sum_{t=1}^T \frac{1}{t} \right) \\
\leq \frac{1}{2} \left( \| x^{(1)} - \hat{x} \|_2^2 + \beta^2 (1 + \log T) \right)
\]

where we used the fact that \( \sum_{t=1}^T 1/t \leq 1 + \log T \). By Jensen’s inequality and convexity of \( f \), we have:

\[
f(\bar{x}(T)) - f(\hat{x}) = f \left( \frac{\sum_{t=1}^T \eta_t x^{(t)}}{\sum_{t=1}^T \eta_t} \right) - f(\hat{x}) \\
\leq \frac{\sum_{t=1}^T \eta_t f(x^{(t)})}{\sum_{t=1}^T \eta_t} - f(\hat{x}) \\
= \frac{\sum_{t=1}^T \eta_t (f(x^{(t)}) - f(\hat{x}))}{\sum_{t=1}^T \eta_t} \\
= \frac{\sum_{t=1}^T \eta_t (f(x^{(t)}) - f(\hat{x}))}{\sum_{t=1}^T \eta_t} \cdot \frac{1}{\sqrt{T}} \\
\leq \frac{1}{2} \left( \| x^{(1)} - \hat{x} \|_2^2 + \beta^2 (1 + \log T) \right) \cdot \frac{2 \beta}{K (\sqrt{T} - 1)}
\]

where we used the fact that \( 2(\sqrt{T} - 1) \leq \sum_{t=1}^T 1/\sqrt{t} \). This proves our claim. \( \square \)
3 A more general setting?

Now, consider the following optimization problem where \( f, r : \mathbb{R}^p \to \mathbb{R} \) are convex and \( K \)-Lipschitz with respect to the \( \ell_2 \)-norm:

\[
\hat{x} = \arg \min_{x \in \mathbb{R}^p} f(x) + r(x)
\]  

Consider the next subgradient descent algorithm for solving the above problem:

**Algorithm 2** Subgradient descent algorithm

<table>
<thead>
<tr>
<th>Input:</th>
<th>Number of iterations ( T \geq 1 ), factor ( \beta &gt; 0 ), initial point ( x^{(1)} \in \mathbb{R}^p ) (The setting of ( x^{(1)} ) can be uninformed, e.g., ( x^{(1)} = 0 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>for ( t = 1 \ldots T - 1 ) do</td>
<td>( x^{(t+1/2)} \leftarrow x^{(t)} - \eta_t g^{(t)} ) where ( g^{(t)} \in \partial f(x^{(t)}) )</td>
</tr>
<tr>
<td></td>
<td>( x^{(t+1)} \leftarrow \arg \min_{x \in \mathbb{R}^p} \left( \frac{1}{2} | x - x^{(t+1/2)} |^2 + \eta_{t+1} r(x) \right) )</td>
</tr>
<tr>
<td>end for</td>
<td></td>
</tr>
<tr>
<td>Output:</td>
<td>( \tilde{x}^{(T)} \leftarrow \frac{\sum_{t=1}^{T} \eta_t x^{(t)}}{\sum_{t=1}^{T} \eta_t} )</td>
</tr>
</tbody>
</table>

Before going into more general observations, let's first consider an example in order to show the usefulness of the above.

**Sparse optimization example.** Let \( r(x) = \lambda \| x \|_1 \). In this case, the second assignment reduces to \( p \) independent assignments of the form:

\[
(\forall j = 1, \ldots, p) \quad x_j^{(t+1)} \leftarrow \arg \min_{x_j \in \mathbb{R}} \left( \frac{1}{2} (x_j - x_j^{(t+1/2)})^2 + \eta_{t+1} \lambda |x_j| \right)
\]

\[
\leftarrow \text{sgn} \left( x_j^{(t+1/2)} \right) \max \left( 0, |x_j^{(t+1/2)}| - \eta_{t+1} \lambda \right)
\]

While the Algorithm 2 might look like a drastic generalization of Algorithm 1, in fact it is not. Note that:

\[
x^{(t+1)} = \arg \min_{x \in \mathbb{R}^p} \left( \frac{1}{2} \| x - x^{(t+1/2)} \|^2 + \eta_{t+1} r(x) \right)
\]  

Recall that \( g^{(t)} \in \partial f(x^{(t)}) \) and let \( s^{(t)} \in \partial r(x^{(t)}) \). Note that \( x^{(t+1)} \) is optimal if and only if \( 0 \) belongs to the subdifferential set of eq.(3) evaluated at \( x^{(t+1)} \):

\[
0 \in \partial \left( \frac{1}{2} \| x - x^{(t+1/2)} \|^2 + \eta_{t+1} r(x) \right) \bigg|_{x = x^{(t+1)}}
\]

\[
0 \in \left( x - x^{(t+1/2)} + \eta_{t+1} \partial r(x) \right) \bigg|_{x = x^{(t+1)}}
\]

\[
0 = x^{(t+1)} - x^{(t+1/2)} + \eta_{t+1} s^{(t+1)}
\]

\[
= x^{(t+1)} - x^{(t)} + \eta_t g^{(t)} + \eta_{t+1} s^{(t+1)}
\]

4
Reorganizing the above, we obtain:

\[ x^{(t+1)} = x^{(t)} - \eta_t g^{(t)} - \eta_{t+1}s^{(t+1)} \]

which looks a lot like a regular subgradient descent step.

4 Exercise

Prove convergence of Algorithm 2.

References