Proximal Gradient and Subgradient Methods

Optimisation for Non Differentiable Functions

Dhruman Gupta

December 16, 2025

Proximal Gradient and Subgradient Methods

- We want to minimize convex functions
- When these are differentiable, we can use gradient descent. For an error of ϵ, we need $O(1/\epsilon)$ iterations.
- What if f is not differentiable?

Subgradient Method

The subgradient method allows us to minimize convex functions that are not differentiable.

Definition: Subgradient

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. A subgradient of f at x is a vector g such that $f(y) \geq f(x) + g^T(y - x)$ for all y.

Note: f is differentiable at x if and only if the subgradient $\partial f(x) = \{\nabla f(x)\}$.

Subgradients

We say f is subdifferentiable at x if the subgradient $\partial f(x)$ is non-empty.

Note that:

$$\partial f(x) = \bigcap_{z \in \textbf{dom } f} \{g | f(z) \geq f(x) + g^T(z - x)\}$$

So $\partial f(x)$ is an infinite intersection of closed half-spaces, and therefore closed and convex.

Existence of Subgradients

If f is convex, and $x \in \textbf{int dom } f$, then $\partial f(x)$ is non-empty.

To prove this, let's first define the supporting hyperplane and the supporting hyperplane theorem.

Supporting Hyperplane

A hyperplane H supports a set $S \subseteq \mathbb{R}^n$ at $x \in S$ if:

1. S is contained in one of the halfspaces defined by the hyperplane.
2. The hyperplane touches S at x.

Supporting Hyperplane Theorem

Supporting Hyperplane Theorem

Let S be a convex set, and let $x \in \textbf{bdry} S$. Then there exists a hyperplane H that supports S at x.

We will take this for granted. Now, let's prove the existence of subgradients.

Proof of Existence of Subgradients

epi $f = \{(x, t) | x \in \mathbb{R}^n, t \in \mathbb{R}, f(x) \leq t\}$. It is convex when f is convex.

Now, take $(x, f(x))$. This is in the epigraph of f. Applying the supporting hyperplane theorem, there exists a hyperplane H that supports **epi** f at $(x, f(x))$.

So $\exists a \in \mathbb{R}^n, b \in \mathbb{R}$ such that $\forall (z, t) \in$ **epi** f:

$$a^T(z - x) + b(t - f(x)) \leq 0$$

Proof of Existence of Subgradients

$$a^T(z - x) + b(t - f(x)) \leq 0$$

This is true $\forall (z, t) \in \textbf{epi } f$. Take $t \to \infty$.
Thus, we must have $b \leq 0$

Case #1: $b < 0$: Divide both sides by b:

$$\frac{a}{b}^T(z - x) + (t - f(x)) \geq 0$$
$$f(z) \geq f(x) + \frac{-a}{b}^T(z - x)$$

So, $g = \frac{-a}{b}$ is a subgradient of f at x.

Proof of Existence of Subgradients

Case $\#2$: $b = 0$: Now, our equation becomes:

$$a^T(z - x) \leq 0 \forall z \in \textbf{dom } f$$

As x is in the interior of the domain, $\exists \epsilon > 0, d \in \mathbb{R}^n$ s.t $x + \epsilon d \in \textbf{dom } f$ and $x - \epsilon d \in \textbf{dom } f$.

$$a^T \epsilon d \leq 0, \qquad a^T(-\epsilon d) \leq 0$$

So, $a = 0$. But, if a, b are both zero, then the hyperplane is the entire space, and thus is not a valid supporting hyperplane.

This is a contradiction to the supporting hyperplane theorem. Thus, $b \neq 0$.

The Subgradient Method

So what is the subgradient method? It is given by:

$$x_{k+1} = x_k - t_k g_k$$

where $g_k \in \partial f(x_k)$.

If f is differentiable, then this is just gradient descent.

Since we do not always descend, we keep track of the best value:

$$f(x_{best}^{(k)}) = \min_{i=1}^{k} f(x_i)$$

Choosing the Step Size

In gradient descent, we can adaptively choose the step size, often done in optimisers like Adam.

However, for subgradient methods, we can't do this, because we don't have "the" gradient. So we choose a fixed step size, or step sizes t_k such that:

$$\sum_{k=1}^{K} t_k = \infty, \qquad \sum_{k=1}^{K} t_k^2 < \infty$$

Convergence of the Subgradient Method

Assume f is convex and G-Lipschitz continuous. Then we have:

Fixed Step Size

$$\lim_{k \to \infty} f(x_{best}^{(k)}) = f^* + \frac{G^2 t}{2}$$

Converging Step Size

$$\lim_{k \to \infty} f(x_{best}^{(k)}) = f^*$$

Proof of Convergence

$$
\begin{aligned}
&\left\|x^k - x^*\right\|_2^2 \\
&= \left\|x^{k-1} - t_k g^{k-1} - x^*\right\|_2^2 \\
&= \left\|x^{k-1} - x^*\right\|_2^2 + t_k^2 \left\|g^{k-1}\right\|_2^2 - 2t_k < g^{k-1}, x^{k-1} - x^* >
\end{aligned}
$$

g is a subgradient of f at x^{k-1}, so:

$$
\begin{aligned}
&f(x^*) \geq f(x^{k-1}) + < g, x^* - x^{k-1} > \\
&\quad \implies < g, x^{k-1} - x^* > \geq f(x^{k-1}) - f(x^*) \\
&\quad \implies -2t_k < g, x^{k-1} - x^* > \leq -2t_k(f(x^{k-1}) - f(x^*))
\end{aligned}
$$

So:

$$\left\|x^k - x^*\right\|_2^2 \leq \left\|x^{k-1} - x^*\right\|_2^2 + t_k^2 \left\|g^{k-1}\right\|_2^2 - 2t_k(f(x^{k-1}) - f(x^*))$$

Proof of Convergence

$$\left\|x^k - x^*\right\|_2^2 \leq \left\|x^{k-1} - x^*\right\|_2^2 + t_k^2 \left\|g^{k-1}\right\|_2^2 - 2t_k(f(x^{k-1}) - f(x^*))$$

Iterating this, we get:

$$\left\|x^k - x^*\right\|_2^2 \leq \|x^0 - x^*\|_2^2 + \sum_{i=1}^{k} t_i^2 \left\|g^{i-1}\right\|_2^2 - 2\sum_{i=1}^{k} t_i(f(x^{i-1}) - f(x^*))$$

Set $R = \left\|x^0 - x^*\right\|_2$. We know that $\left\|x^k - x^*\right\|_2^2 \geq 0$. So:

$\sum_{i=1}^{k} t_i(f(x^{i-1}) - f(x^*)) \leq \frac{R^2 + \sum_{i=1}^{k} t_i^2 \left\|g^{i-1}\right\|_2^2}{2}$

Proof of Convergence

$\sum_{i=1}^{k} t_i(f(x^{i-1}) - f(x^*)) \leq \frac{R^2 + \sum_{i=1}^{k} t_i^2 \left\|g^{i-1}\right\|_2^2}{2}$

Now:

$$\left(f(x_{best}^{(k)}) - f(x^*)\right) \sum t_i \leq \sum_{i=1}^{k} t_i(f(x^{i-1}) - f(x^*))$$

$$f(x_{best}^{(k)}) - f(x^*) \leq \frac{R^2 + G^2 \sum_{i=1}^{k} t_i^2}{2 \sum t_i}$$

For fixed step size t, this becomes:

$$f(x_{best}^{(k)}) - f(x^*) \leq \frac{R^2}{2kt} + \frac{G^2 t}{2}$$

Taking limit proves first claim.

Proof of Convergence

$$f(x_{best}^{(k)}) - f(x^*) \leq \frac{R^2 + G^2 \sum_{i=1}^{k} t_i^2}{2 \sum t_i}$$

For dynamic step size t_k, the numerator is finite and the denominator is infinite as $k \to \infty$. So:

$$\lim_{k \to \infty} f(x_{best}^{(k)}) = f(x^*)$$

For convergence analysis, say we want error ϵ. Then we can set:

$\frac{R^2}{2kt} = \frac{G^2 t}{2} = \frac{\epsilon}{2}$. Then, $t = \frac{\epsilon}{G^2}$, and $k = \frac{R^2 G^2}{\epsilon^2}$.
R, G constants, so $k = O(\frac{1}{\epsilon^2})$.

Can we do better?

$$k = O(\frac{1}{\epsilon^2})$$

This is very bad compared to gradient descent, which only needs $O(1/\epsilon)$ iterations.

Can we do better?

In this setting, it can be shown that for any starting point and time step size, there always exists a function f where this method will take $O(1/\epsilon^2)$ iterations.

So, let's see a setting (popular) where we can do better.

Composite Functions

Let $f(x) = g(x) + h(x)$, where g is convex and differentiable, and h is convex and non-differentiable.

We want to minimize $f(x)$. Can we somehow combine gradient descent and subgradient method to get a better convergence rate?

Recall that for gradient descent, if ∇f is L-Lipschitz, then we are minimising:

$$x_{next} = \arg\min_{z} \left(f(x) + \nabla f(x)^T (z - x) + \frac{1}{2t} \|z - x\|_2^2 \right)$$

Majorization (Descent Lemma)

Recall that for gradient descent, if ∇f L-Lipschitz, then we are minimising:

$$x_{next} = \arg\min_z \left(f(x) + \nabla f(x)^T (z - x) + \frac{1}{2t} \|z - x\|_2^2 \right)$$

The reason is that for such f:

$$f(z) \leq f(x) + \nabla f(x)^T (z - x) + \frac{1}{2t} \|z - x\|_2^2$$

when $t \leq \frac{1}{L}$ (this is called the descent lemma).

Intuition for Proximal Gradient Descent

$$x_{next} = \arg\min_z \left(f(x) + \nabla f(x)^T (z - x) + \frac{1}{2t} \|z - x\|_2^2 \right)$$

If f is differentiable, then to solution to this is $x_{next} = x - t\nabla f(x)$ - exactly the gradient descent step.

But now f is not differentiable. So why don't we minimise g using this method, and leave h alone?

Proximal Function

So:

$$\begin{aligned}&\arg\min_z \left(g(x) + \nabla g(x)^T (z-x) + \frac{1}{2t}\left\|z-x\right\|_2^2 + h(z)\right)\\ &= \arg\min_z \frac{1}{2t}\left\|z-(x-t\nabla g(x))\right\|_2^2 + h(z)\end{aligned}$$

This is exactly the proximal gradient descent step. Define:

$prox_{h,t} = \arg\min_z \frac{1}{2t}\left\|z-x\right\|_2^2 + h(z)$

Proximal Gradient Descent

$prox_{h,t}(x) = \arg\min_z \frac{1}{2t} \|z - x\|_2^2 + h(z)$

The gradient descent step is:

$$x^k = prox_{h,t_k}(x^{k-1} - t_k \nabla g(x^{k-1}))$$

Example: LASSO Loss

Let X be data points (inputs), and y be labels. For a linear model, our least squares loss is:

$$\frac{1}{2}\left\|X\beta - y\right\|_2^2$$

where β are the weights. In regularization, we want to penalise the magnitude of β. Say we use the L_1 norm. I.e:

$$f(\beta) = \frac{1}{2}\left\|X\beta - y\right\|_2^2 + \lambda\left\|\beta\right\|_1$$

This is the LASSO loss. Note that the first term is differentiable, and the second term is not.

Proximal Gradient Descent for LASSO

The proximal gradient descent step is:

$$\beta^k = prox_{\lambda\|\beta\|_1, t_k}(\beta^{k-1} + t_k X^T(y - X\beta^{k-1}))$$

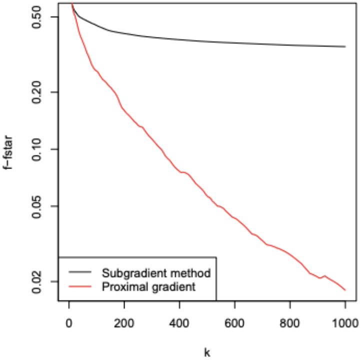
Here:

$$[prox_{\lambda\|\beta\|_1, t_k}]_i = \begin{cases} \beta_i - \lambda, & \text{if } \beta_i > \lambda, \\ 0, & \text{if } -\lambda \leq \beta_i \leq \lambda, \quad i = 1, \ldots, n, \\ \beta_i + \lambda, & \text{if } \beta_i < -\lambda \end{cases}$$

This is called the iterative soft thresholding algorithm (ISTA).

Proximal Gradient Descent for LASSO

Example of proximal gradient (ISTA) vs. subgradient method convergence curves



Practically used in all LASSO losses.

Convergence Analysis

If

1. g is convex, differentiable, with domain $\mathbb{R}^n$, and ∇g is L-Lipschitz.
2. h is convex and $prox_{h,t}$ can be computed

Then, the convergence rate is $O(1/\epsilon)$.

So we do much better than the subgradient method!

Note: if $prox_{h,t}$ is expensive to compute, then each step takes much longer and convergence is costly. But for many problems, we know a good closed form.

Bibliography i

- Boyd, S et al. (2022). **Subgradients.** URL: https://web.stanford.edu/class/ee364b/lectures/subgradients_notes.pdf.
- Tibshirani, Ryan (n.d.[a]). **Proximal Gradient Descent (and Acceleration).** URL: https://www.stat.cmu.edu/~ryantibs/convexopt/lectures/prox-grad.pdf (visited on 12/16/2025).
- — (n.d.[b]). **Subgradient Method.** URL: https://www.stat.cmu.edu/~ryantibs/convexopt/lectures/sg-method.pdf (visited on 12/16/2025).

Thank you!