CSED490Y: Optimization for Machine Learning

Week 07-1: Proximal gradient descent

Namhoon Lee

POSTECH

Spring 2022

Projected gradient method

Constrained minimization problems:

$$\min_{x \in \mathbb{C}} f(x)$$

Projected gradient method

Constrained minimization problems:

$$\min_{x \in \mathbb{C}} f(x)$$

Projected gradient method:

$$x_{t+1} = \text{proj}_{\mathbb{C}}(x_t - \eta \nabla f(x_t))$$

where $\text{proj}_{\mathbb{C}}(\cdot)$ is the projection operation defined as

$$\text{proj}_{\mathbb{C}}(x_0) = \arg\min_{x \in \mathbb{C}} \frac{1}{2}\|x - x_0\|_2^2$$

Projected gradient method

Constrained minimization problems:

$$\min_{x \in \mathbb{C}} f(x)$$

Projected gradient method:

$$x_{t+1} = \mathrm{proj}_{\mathbb{C}}(x_t - \eta \nabla f(x_t))$$

where $\mathrm{proj}_{\mathbb{C}}(\cdot)$ is the projection operation defined as

$$\mathrm{proj}_{\mathbb{C}}(x_0) = \arg\min_{x \in \mathbb{C}} \frac{1}{2} \|x - x_0\|_2^2$$

- Same convergence rates as gradient method; *e.g.* $\mathcal{O}(1/\epsilon^2)$ for convex and Lipschitz continuous functions.

Projected gradient method

An equivalent formulation to constrained minimization:

$$\min_{x \in \mathbb{C}} f(x) \quad \equiv \quad \min_{x} f(x) + \mathcal{I}_{\mathbb{C}}(x)$$

where $\mathcal{I}_{\mathbb{C}}$ is an indicator function

$$\mathcal{I}_{\mathbb{C}} = \begin{cases} 0 & \text{if } x \in \mathbb{C} \\ \infty & \text{if } x \notin \mathbb{C} \end{cases}$$

which is simple to evaluate and convex if $\mathbb{C}$ is convex (but not smooth).

Projected gradient method

An equivalent formulation to constrained minimization:

$$\min_{x \in \mathbb{C}} f(x) \quad \equiv \quad \min_{x} f(x) + \mathcal{I}_{\mathbb{C}}(x)$$

where $\mathcal{I}_{\mathbb{C}}$ is an indicator function

$$\mathcal{I}_{\mathbb{C}} = \begin{cases} 0 & \text{if } x \in \mathbb{C} \\ \infty & \text{if } x \notin \mathbb{C} \end{cases}$$

which is simple to evaluate and convex if $\mathbb{C}$ is convex (but not smooth).

This penalty form can be applied to the projection operator, *i.e.*

$$\begin{aligned} \mathrm{proj}_{\mathbb{C}}(x_0) &= \arg\min_{x \in \mathbb{C}} \frac{1}{2}\|x - x_0\|_2^2 \\ &= \arg\min_{x} \frac{1}{2}\|x - x_0\|_2^2 + \mathcal{I}_{\mathbb{C}}(x) \qquad \checkmark \end{aligned}$$

Composite functions

Consider f as a composite function of g and h:

$$f(x) = g(x) + h(x)$$

- g is convex and differentiable.
- h is convex, but not necessarily differentiable.

Composite functions

Consider f as a composite function of g and h:

$$f(x) = g(x) + h(x)$$

f - G-Lipschitz

subgrad converges

$O(1/\sqrt{T}) \Longleftrightarrow O(1/\varepsilon^2)$

- g is convex and differentiable.
- h is convex, but not necessarily differentiable.

If f were differentiable we could apply gradient descent; yet only g is differentiable.

- Subgradient method? Can we could do better?

Interpretation for proximal gradient

Recall that the gradient descent algorithm can be interpreted as minimizing a quadratic approximation:

$$x_{t+1} = \arg\min_x f(x) \approx f(x_t) + \langle \nabla f(x_t), x - x_t \rangle + \frac{1}{2\eta}\|x - x_t\|_2^2$$

i.e., taking derivative and solving it w.r.t. x will give gradient descent.

Interpretation for proximal gradient

Recall that the gradient descent algorithm can be interpreted as minimizing a quadratic approximation:

$$x_{t+1} = \arg\min_x f(x) \approx f(x_t) + \langle \nabla f(x_t), x - x_t \rangle + \frac{1}{2\eta}\|x - x_t\|_2^2$$

i.e., taking derivative and solving it w.r.t. x will give gradient descent.

We can do the same for g in the composite function, *i.e.*

$$\begin{aligned} x^+ &= \arg\min_x f(x) \approx \tilde{g}(x) + h(x) \\ &= \arg\min_x \underbrace{g(y) + \langle \nabla g(y), x - y \rangle + \frac{1}{2\eta}\|x - y\|_2^2} + h(x) \\ &= \arg\min_x \frac{1}{2\eta}\|x - \underbrace{(y - \eta\nabla g(y))}\|_2^2 + h(x) \end{aligned}$$

Interpretation for proximal gradient $x^+ = \text{proj}_{\mathbb{C}}(y) = \arg\min_x \frac{1}{2}\|x-y\|_2^2 + \mathcal{I}_{\mathbb{C}}(x)$

Recall that the gradient descent algorithm can be interpreted as minimizing a quadratic approximation:

$$x_{t+1} = \arg\min_x f(x) \approx f(x_t) + \langle \nabla f(x_t), x - x_t \rangle + \frac{1}{2\eta}\|x - x_t\|_2^2$$

i.e., taking derivative and solving it w.r.t. x will give gradient descent.

We can do the same for g in the composite function, *i.e.*

$$\begin{aligned}
x^+ &= \arg\min_x f(x) \approx \tilde{g}(x) + h(x) \\
&= \arg\min_x g(y) + \langle \nabla g(y), x - y \rangle + \frac{1}{2\eta}\|x - y\|_2^2 + h(x) \\
&= \arg\min_x \frac{1}{2\eta}\|x - (y - \eta\nabla g(y))\|_2^2 + h(x)
\end{aligned}$$

This resembles the projection operator except that we now have $h(x)$ instead of $\mathcal{I}_{\mathbb{C}}(x)$.

Proximal operator

Idea: generalize $\mathcal{I}$ to other (convex) functions other than just indicator function.

Proximal operator

Idea: generalize $\mathcal{I}$ to other (convex) functions other than just indicator function.

In general, the proximal operator can be written as follows:

$$\text{prox}_h(y) = \arg\min_x \frac{1}{2}\|x - y\|_2^2 + h(x)$$

i.e., given y try to find x that minimizes $h(x)$, but also don't go too far from y.

Proximal operator

Idea: generalize $\mathcal{I}$ to other (convex) functions other than just indicator function.

In general, the proximal operator can be written as follows:

$$\mathrm{prox}_h(y) = \arg\min_x \frac{1}{2}\|x - y\|_2^2 + h(x)$$

i.e., given y try to find x that minimizes $h(x)$, but also don't go too far from y.

A modification:

$$\mathrm{prox}_{\eta h}(y) = \arg\min_x \frac{1}{2\eta}\|x - y\|_2^2 + h(x)$$

- η small: 1st term explodes, stay close to y (small step size).
- η large: 1st term vanishes, minimize h is what you care (big step size).

Proximal operator evquivalence

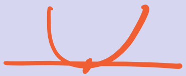
From

$$\mathrm{prox}_h(y) = \arg\min_x \frac{1}{2}\|x - y\|_2^2 + h(x)$$

Update h to ηh

$$\begin{aligned}
\mathrm{prox}_{\eta h}(y) &= \arg\min_x \frac{1}{2}\|x - y\|_2^2 + \eta h(x) \\
&= \arg\min_x \cancel{\eta}\left(\frac{1}{2\eta}\|x - y\|_2^2 + h(x)\right) \\
&= \arg\min_x \frac{1}{2\eta}\|x - y\|_2^2 + h(x)
\end{aligned}$$

Example of prox operator



$0 \in \frac{1}{\eta}(x-y) + \partial\|x\|_1$

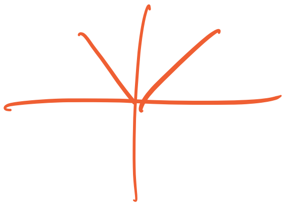
For $h(x) = \|x\|_1$, the proximal operator becomes

$$x^+ = \mathrm{prox}_{\eta h}(y) = \arg\min_x \frac{1}{2\eta}\|x - y\|_2^2 + \|x\|_1$$

$$\partial\|x\|_1 = \begin{cases} -1 & x_i < 0 \\ [-1,1] & x_i = 0 \\ +1 & x_i > 0 \end{cases}$$

where x^+ is "soft-threshold"-ed y

$$x^+ : x_i = \begin{cases} y_i + \eta & \text{for } y_i < -\eta \\ 0 & \text{for } |y_i| \le \eta \\ y_i - \eta & \text{for } y_i > \eta \ . \end{cases}$$



(sol'n) Use the prox operator definition and suboptimality condition for subgradient.

Example of prox operator

For $h(x) = \|x\|_1$, the proximal operator becomes

$$x^+ = \mathrm{prox}_{\eta h}(y) = \arg\min_x \frac{1}{2\eta}\|x - y\|_2^2 + \|x\|_1$$

where x^+ is "soft-threshold"-ed y

$$x^+ : x_i = \begin{cases} y_i + \eta & \text{for } y_i < -\eta \\ 0 & \text{for } |y_i| \le \eta \\ y_i - \eta & \text{for } y_i > \eta \ . \end{cases}$$

(sol'n) Use the prox operator definition and suboptimality condition for subgradient.

"soft-thresholdg"

Exercise: $prox_h((3, -0.7, -2)^\top) = \underbrace{(2, 0, -1)}$.

$\eta = 1$

Proximal gradient method

$y_{t+1} = x_t - \eta \nabla f(x_t)$

$x_{t+1} = \mathrm{proj}_{C}(y_{t+1})$

Proximal gradient:

$$
\begin{aligned}
x_{t+1} &= \mathrm{prox}_{\eta h}(\underbrace{x_t - \eta \nabla g(x_t)}_{y_{t+1}}) \\
&= \arg\min_{x} \frac{1}{2\eta} \|x - (\underbrace{x_t - \eta \nabla g(x_t)}_{y_{t+1}})\|_2^2 + h(x)\ .
\end{aligned}
$$

- If h is indicator function, the proximal gradient is the same as the projected gradient.

Gradient mapping

Define gradient mapping:

$$G_\eta(x) = \frac{1}{\eta}(x - \mathrm{prox}_{\eta h}(x - \eta \nabla g(x))) \ .$$

Then we can rewrite the proximal gradient method into something that looks more like a gradient descent update step:

$$x_{t+1} = x_t - \eta G_\eta(x_t) \ .$$

- G_η is called the gradient map of proximal gradient method, and we treat this as if it's a gradient, but G_η is not a (sub)gradient of f in general.
- We do this to make analyzing convergence behavior easier.

Thank you

Any questions?

Credits

A lot of material in this course is borrowed or derived from the following:

- Numerical Optimization, Jorge Nocedal and Stephen J. Wright.
- Convex Optimization, Stephen Boyd and Lieven Vandenberghe.
- Convex Optimization, Ryan Tibshirani.
- Optimization for Machine Learning, Martin Jaggi and Nicolas Flammarion.
- Optimization Algorithms, Constantine Caramanis.
- Advanced Machine Learning, Mark Schmidt.