Proximal Gradient Method

SIE 449/549: Optimization for Machine Learning

Afrooz Jalilzadeh

Department of Systems and Industrial Engineering
University of Arizona

Gradient Method

- Consider the following optimization problem

$$\min_x \; f(x)$$

  where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and **differentiable**

- Gradient descent: choose initial $x_0 \in \mathbb{R}^n$, repeat:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

  If $\nabla f(x)$ is Lipschitz, gradient descent has convergence rate $\mathcal{O}(1/\epsilon)$ with stepsize $\alpha = 1/L$

- What if f is not differentiable?

Subgradient Method

- Replacing gradients with subgradients: choose initial $x_0 \in \mathbb{R}^n$, repeat:

$$x_{k+1} = x_k - \alpha_k g_k$$

  where $g_k \in \partial f(x_k)$, any subgradient of f at x_k

- **Constant stepsize**: $\alpha_k = \alpha > 0$
- **Diminishing stepsize**: α_k satisfies the following two conditions

$$\lim_{k\to\infty} \alpha_k = 0, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty$$

- Subgradient method is not necessarily a descent method
- Thus, the best solution among all of the iterations is used as the final solution:

$$f(x_k^{best}) = \min_{i=0,\ldots,k} f(x_i)$$

- Subgradient method has convergence rate $\mathcal{O}(1/\epsilon^2)$ with stepsize $\alpha = \epsilon/L^2$, where f is Lipschitz continuous with constant L

Can we do better?

Can we do better than $\mathcal{O}(1/\epsilon^2)$ for convex and non-differentiable functions?

- Yes, if the objective is decomposable into two functions in the following manner:

$$\min f(x) \triangleq g(x) + h(x),$$

  - g is a convex and differentiable function
  - h is convex and possibly non-differentiable, but simple, e.g., $h(x) = \|x\|_1$
  - With the proximal gradient descent method, we can achieve a convergence rate of $\mathcal{O}(1/\epsilon)$

Proximal Gradient Descent

- Simple gradient descent works with a convex and differentiable f, using gradient information to take steps towards the optima
- This step is derived using a quadratic approximation of the objective function $f(x)$, after replacing $\nabla^2 f$ with a spherical term $\frac{1}{\alpha} I$:

$$x_{k+1} = \text{argmin}_z \left\{ f(x_k) + \nabla f(x_k)^T (z - x_k) + \frac{1}{2\alpha} \|z - x_k\|^2 \right\}$$

- If f is not differentiable, but is decomposable into two convex functions g and h, we can still use a quadratic approximation of the smooth part g to define a step towards the minimum value

$$x_{k+1} =$$

$$x_{k+1} =$$

Proximal Gradient Descent

- *prox* is a function of h and α, and is referred to as the proximal map of h:

$$prox_{h,\alpha}(x) = \text{argmin}_z \frac{1}{2\alpha}\|z - x\|^2 + h(z)$$

- Proximal gradient descent can be defined as follows:
  - Choose initial x_0 and then repeat:

$$x_{k+1} = prox_{h,\alpha_k}(x_k - \alpha_k \nabla g(x_k))$$

- To make the update look familiar, we can define the update as follows

$$x_{k+1} = x_k - \alpha_k G_{h,\alpha_k}(x_k),$$

where $G_{h,\alpha}(x) = \dfrac{x - prox_{h,\alpha}(x - \alpha\nabla g(x))}{\alpha}$

Proximal Gradient Descent

- Did we just swapped one minimization problem for another?
  - *prox(.)* is can be computed analytically for a lot of important functions h
  - $prox(.)$ doesn't depend on g at all, only on h
  - Smooth part g can be complicated, we only need to compute its gradients

$h(x)$	$prox_{h,\alpha}(x)$	Assumptions
$\lambda\|x\|$	$\left(1 - \frac{\alpha\lambda}{\max\{\|x\|, \alpha\lambda\}}\right) x$	$\lambda > 0$
$\lambda\|x\|^3$	$\frac{2}{1 + \sqrt{1 + 12\alpha\lambda\|x\|}} x$	$\lambda > 0$
$\lambda\|x\|_1$	$[\lvert x\rvert - \alpha\lambda e]_+ \odot \mathrm{sgn}(x)$	$\lambda > 0$

Table 1: Prox Computation

Properties of Proximal Map

- Postcomposition: $g(x) = \alpha f(x) + b$, with $\alpha > 0$
- Precomposition: $g(x) = f(\alpha x + b)$ with $\alpha \neq 0$
- Seperability: $g(x) = \sum_{i=1}^{n} f_i(x_i)$, where $x = [x_i]_{i=1}^{n}$
- Affine addition: $g(x) = f(x) + a^T x + b$
- Nonexpansivity:

Convergence Analysis

- Consider the following problem:

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

- The function g is convex, differentiable, $dom(g) = \mathbb{R}^n$, and ∇g is Lipschitz continuous with L
- The function h is convex and its proximal map can be easily computed
- Proximal gradient descent with fixed step size $\alpha \leq 1/L$ satisfies:

$$f(x_k) - f(x^*) \leq \frac{\|x_0 - x^*\|}{2\alpha k}$$

- Proximal gradient descent has a convergence rate of ....

Lasso

- Consider data points (a_i, b_i), $i = 1, \ldots, m$, where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$
- Suppose $A \in \mathbb{R}^{m \times n}$ denote the predictor matrix (whose i^{th} row is a_i) and b denote the response vector
- Least square problem is formulated as:
- *Least absolute selection* and *shrinkage operator* or **lasso**, is defined as:

  where $\lambda \geq 0$ is tuning parameter
- Why Lasso?
- Why care about sparsity?
- Larger values of the tuning parameter λ typically means sparser solutions

Proximal Gradient Method to Solve Lasso

- Solve Lasso problem using proximal gradient method:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1$$