Optimization for Machine Learning CS-439

Lecture 4: Projected, Proximal, Subgradient and Stochastic Gradient Descent

Martin Jaggi

EPFL – github.com/epfml/OptML_course

March 16, 2018

Strongly convex constrained minimization: $\mathcal{O}(\log(1/\varepsilon))$ steps

Theorem

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable. Let $X \subseteq \mathbb{R}^d$ be a closed and convex set and suppose that f is smooth over X with parameter L and strongly convex over X with parameter $\mu > 0$. Choosing

$$\gamma := \frac{1}{L},$$

projected gradient descent with arbitrary $\mathbf{x}_0$ satisfies

(i)

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \leq \left(1 - \frac{\mu}{L}\right) \|\mathbf{x}_t - \mathbf{x}^\star\|^2, \quad t \geq 0.$$

(ii)

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \frac{L}{2}\left(1 - \frac{\mu}{L}\right)^t \|\mathbf{x}_0 - \mathbf{x}^\star\|^2.$$

Strongly convex constrained minimization: $\mathcal{O}(\log(1/\varepsilon))$ steps

Proof.

Strengthen the "constrained" vanilla bound

$$\frac{1}{2\gamma}\left(\gamma^2\|\nabla f(\mathbf{x}_t)\|^2+\|\mathbf{x}_t-\mathbf{x}^\star\|^2-\|\mathbf{x}^+-\mathbf{x}^\star\|^2-\|\mathbf{y}^+-\mathbf{x}^+\|^2\right)$$

to

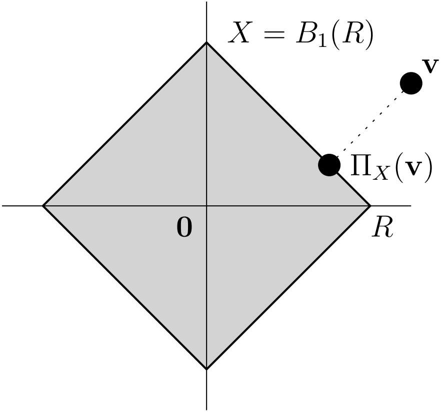
$$\frac{1}{2\gamma}\left(\gamma^2\|\nabla f(\mathbf{x}_t)\|^2+\|\mathbf{x}_t-\mathbf{x}^\star\|^2-\|\mathbf{x}^+-\mathbf{x}^\star\|^2-\|\mathbf{y}^+-\mathbf{x}^+\|^2\right) -\frac{\mu}{2}\|\mathbf{x}_t-\mathbf{x}^\star\|^2$$

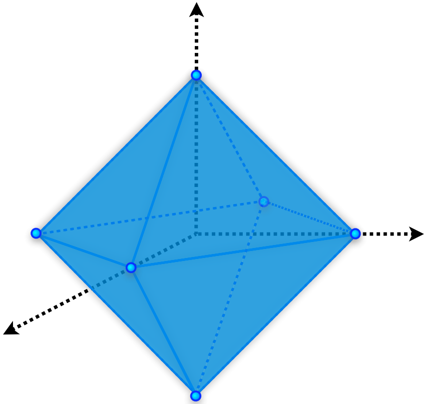
using strong convexity.

Then proceed as in the unconstrained theorem. □

Projecting onto ℓ_1-balls

$$X = B_1(R) := \left\{ \mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_1 = \sum_{i=1}^{d} |x_i| \le R \right\}$$





2^d facets!

Projecting onto ℓ_1-balls

w.l.o.g.

- $R = 1$, (*)
- $v_i \geq 0$ for all i,
- $\sum_{i=1}^{d} v_i > 1$.

And using this,

$\mathbf{x} = \Pi_X(\mathbf{v})$ satisfies $x_i \geq 0$ for all i and $\sum_{i=1}^{d} x_i = 1$.

Projecting onto ℓ_1-balls

Corollary

Under our assumption (*),

$$\Pi_X(\mathbf{v}) = \underset{\mathbf{x}\in\Delta_d}{\operatorname{argmin}} \|\mathbf{x}-\mathbf{v}\|^2,$$

where

$$\Delta_d := \left\{\mathbf{x}\in\mathbb{R}^d : \sum_{i=1}^d x_i = 1, x_i \geq 0\ \forall i\right\}$$

is the standard simplex.

Also, w.l.o.g. assume that v is ordered increasingly, $v_1 \geq v_2 \geq \cdots \geq v_d$.

Projecting onto ℓ_1-balls

Lemma

Let $\mathbf{x}^\star := \operatorname{argmin}_{\mathbf{x}\in\Delta_d} \|\mathbf{x}-\mathbf{v}\|^2$, *and* $\mathbf{v}$ *ordered increasingly. There exists (a unique) index* $p \in \{1,\ldots,d\}$ *s.t.*

$$\begin{aligned} x_i^\star &> 0, \quad i \le p, \\ x_i^\star &= 0, \quad i > p. \end{aligned}$$

Proof.

Optimality criterion for constrained optimization:

$$\nabla d_{\mathbf{v}}(\mathbf{x}^\star)^\top(\mathbf{x}-\mathbf{x}^\star) = 2(\mathbf{x}^\star-\mathbf{v})^\top(\mathbf{x}-\mathbf{x}^\star) \ge 0, \quad \forall \mathbf{x}\in\Delta_d.$$

$\exists$ a positive entry in $\mathbf{x}^\star$ (because $\sum_{i=1}^d x_i^\star = 1$).
Why not $x_i^\star = 0$ and $x_{i+1}^\star > 0$? If so, we could decrease $x_{i+1}^\star$ by ε and increase $x_i^\star$ to ε to obtain $\mathbf{x}\in\Delta_d$ s.t.

$$(\mathbf{x}^\star-\mathbf{v})^\top(\mathbf{x}-\mathbf{x}^\star) = (0-v_i)\varepsilon-(x_{i+1}^\star-v_{i+1})\varepsilon = \varepsilon(\underbrace{v_{i+1}-v_i}_{\le 0}-\underbrace{x_{i+1}^\star}_{>0}) < 0,$$

contradicting the optimality. □

Projecting onto ℓ_1-balls

Can say more about $\mathbf{x}^\star$:

Lemma

With p as in the above Lemma, and $\mathbf{v}$ ordered increasingly, we have

$$x_i^\star = v_i - \Theta_p, \quad i \leq p,$$

where

$$\Theta_p = \frac{1}{p}\Big(\sum_{i=1}^{p} v_i - 1\Big).$$

Proof.

Assume there is $i, j \leq p$ with $x_i^\star - v_i < x_j^\star - v_j$. As before, we could decrease $x_j^\star > 0$ by ε and increase $x_i^\star$ by ε to get $\mathbf{x} \in \Delta_d$ s.t.

$$(\mathbf{x}^\star - \mathbf{v})^\top (\mathbf{x} - \mathbf{x}^\star) = (x_i^\star - v_i)\varepsilon - (x_j^\star - v_j)\varepsilon = \varepsilon\big(\underbrace{(x_i^\star - v_i) - (x_j^\star - v_j)}_{<0}\big) < 0,$$

again contradicting optimality of $\mathbf{x}^\star$. □

Projecting onto ℓ_1-balls

Summary: have d candidates for $\mathbf{x}^\star$, namely

$$\mathbf{x}^\star(p) := (v_1 - \Theta_p, \ldots, v_p - \Theta_p, 0, \ldots, 0), \quad p \in \{1, \ldots, d\},$$

Need to find the right one. In order for candidate $\mathbf{x}^\star(p)$ to comply with our first Lemma, we must have

$$v_p - \Theta_p > 0,$$

and this actually ensures $\mathbf{x}^\star(p)_i > 0$ for all $i \leq p$ (because $\mathbf{v}$ is ordered) and therefore $\mathbf{x}^\star(p) \in \Delta_d$.

But there could still be several choices for p. Among them, we simply pick the one for which $\mathbf{x}^\star(p)$ minimizes the distance to $\mathbf{v}$.

In time $\mathcal{O}(d \log d)$, by first sorting v and checking incrementally.

Projecting onto ℓ_1-balls

Theorem

Let $\mathbf{v} \in \mathbb{R}^d$*,* $R \in \mathbb{R}_+$*,* $X = B_1(R)$ *the* ℓ_1*-ball around* $\mathbf{0}$ *of radius* R*. The projection*

$$\Pi_X(\mathbf{v}) = \operatorname*{argmin}_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{v}\|^2$$

of $\mathbf{v}$ *onto* $B_1(R)$ *can be computed in time* $\mathcal{O}(d \log d)$*.*

This can be improved to time $\mathcal{O}(d)$ by avoiding sorting.

Section 3.6

Proximal Gradient Descent

Composite optimization problems

Consider objective functions composed as

$$f(\mathbf{x}) := g(\mathbf{x}) + h(\mathbf{x})$$

where g is a "nice" function, where as h is a "simple" additional term, which however doesn't satisfy the assumptions of niceness which we used in the convergence analysis so far.

In particular, an important case is when h is not differentiable.

Idea

The classical gradient step for minimizing g:

$$\mathbf{x}_{t+1} = \operatorname*{argmin}_{\mathbf{y}} \; g(\mathbf{x}_t) + \nabla g(\mathbf{x}_t)^\top (\mathbf{y} - \mathbf{x}_t) + \frac{1}{2\gamma} \|\mathbf{y} - \mathbf{x}_t\|^2 \; .$$

For the stepsize $\gamma := \frac{1}{L}$ it exactly minimizes the local quadratic model of g at our current iterate $\mathbf{x}_t$, formed by the smoothness property with parameter L.

Now for $f = g + h$, keep the same for g, and add h unmodified.

$$\begin{aligned} \mathbf{x}_{t+1} :=& \operatorname*{argmin}_{\mathbf{y}} \; g(\mathbf{x}_t) + \nabla g(\mathbf{x}_t)^\top (\mathbf{y} - \mathbf{x}_t) + \frac{1}{2\gamma} \|\mathbf{y} - \mathbf{x}_t\|^2 + h(\mathbf{y}) \\ =& \operatorname*{argmin}_{\mathbf{y}} \; \frac{1}{2\gamma} \|\mathbf{y} - (\mathbf{x}_t - \gamma \nabla g(\mathbf{x}_t))\|^2 + h(\mathbf{y}) \; , \end{aligned}$$

the proximal gradient descent update.

The proximal gradient descent algorithm

An iteration of proximal gradient descent is defined as

$$\mathbf{x}_{t+1} := \mathrm{prox}_{h,\gamma}(\mathbf{x}_t - \gamma \nabla g(\mathbf{x}_t)) \ .$$

Or equivalently

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma G_\gamma(\mathbf{x}_t)$$

for $G_{h,\gamma}(\mathbf{x}) := \frac{1}{\gamma}\Big(\mathbf{x} - \mathrm{prox}_{h,\gamma}(\mathbf{x} - \gamma \nabla g(\mathbf{x}))\Big)$ being the so called generalized gradient of f.

A generalization of gradient descent?

- $h \equiv 0$: recover gradient descent
- $h \equiv \boldsymbol{\iota}_X$: recover projected gradient descent!

  Given a closed convex set X, the indicator function of the set X is given as the convex function

$$\boldsymbol{\iota}_X : \mathbb{R}^d \to \mathbb{R} \cup +\infty$$
$$\mathbf{x} \mapsto \boldsymbol{\iota}_X(\mathbf{x}) := \begin{cases} 0 & \text{if } \mathbf{x} \in X, \\ +\infty & \text{otherwise.} \end{cases}$$

  Proximal mapping becomes

$$\operatorname{prox}_{h,\gamma}(\mathbf{z}) := \underset{\mathbf{y}}{\operatorname{argmin}} \left\{ \frac{1}{2\gamma} \|\mathbf{y}-\mathbf{z}\|^2 + \boldsymbol{\iota}_X(\mathbf{y}) \right\} = \underset{\mathbf{y}\in X}{\operatorname{argmin}} \|\mathbf{y}-\mathbf{z}\|^2$$

Convergence in $\mathcal{O}(1/\varepsilon)$ steps

Same as vanilla case for smooth functions, but now for any h for which we can compute the proximal mapping.

Chapter 4

Subgradient Descent

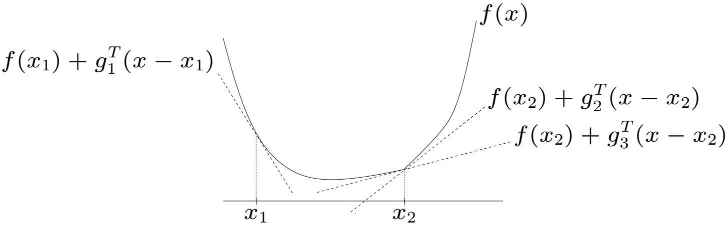
Subgradients

What if f is not differentiable?

Definition

$\mathbf{g} \in \mathbb{R}^d$ is a subgradient of f at $\mathbf{x}$ if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{g}^\top(\mathbf{y} - \mathbf{x}) \qquad \text{for all } \mathbf{y} \in \mathbf{dom}(f)$$



And: $\partial f(\mathbf{x}) \subseteq \mathbb{R}^d$ is the set of subgradients of f at $\mathbf{x}$.

What are subgradients good for?

Convexity

Lemma (Exercise 22)

A function $f : \mathbf{dom}(f) \to \mathbb{R}$ is convex if and only if $\mathbf{dom}(f)$ is convex and $\partial f(\mathbf{x}) \neq \emptyset$ for all $\mathbf{x} \in \mathbf{dom}(f)$.

Lipschitz Continuity

Lemma (Exercise 24)

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex, $B \in \mathbb{R}_+$. Then the following two statements are equivalent.

(i) $\|\mathbf{g}\| \leq B$ *for all* $\mathbf{x} \in \mathbb{R}^d$ *and all* $\mathbf{g} \in \partial f(\mathbf{x})$.

(ii) $|f(\mathbf{x}) - f(\mathbf{y})| \leq B\|\mathbf{x} - \mathbf{y}\|$ *for all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

What are subgradients good for?

Subgradient Optimality Condition. Subgradients also allow us to describe cases of optimality for functions which are not necessarily differentiable (and not necessarily convex)

Lemma

Suppose that f is any function over $\mathbf{dom}(f)$, and $\mathbf{x} \in \mathbf{dom}(f)$. If $\mathbf{0} \in \partial f(\mathbf{x})$, then $\mathbf{x}$ is a global minimum.

Proof.

□

The subgradient descent algorithm

An iteration of subgradient descent is defined as

$$\text{Let } \mathbf{g}_t \in \partial f(\mathbf{x}_t)$$
$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma \mathbf{g}_t.$$

Bounded subgradients: $\mathcal{O}(1/\varepsilon^2)$ steps

The following result gives the convergence for Subgradient Descent. It is identical to Theorem 2.1, up to relaxing the requirement of differentiability.

Theorem

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and B-Lipschitz continuous on $\mathbb{R}^d$ with a global minimum $\mathbf{x}^\star$; furthermore, suppose that $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$. Choosing the constant stepsize

$$\gamma := \frac{R}{B\sqrt{T}},$$

subgradient descent yields

$$\frac{1}{T}\sum_{t=0}^{T-1} f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \frac{RB}{\sqrt{T}}.$$

Bounded subgradients: $\mathcal{O}(1/\varepsilon^2)$ steps

Proof.

$\square$

Optimality of first-order methods

With all the convergence rates we have seen so far, a very natural question to ask is if these rates are best possible or not.
Surprisingly, the rate can indeed not be improved in general.

Theorem (Nesterov)

For any $T \leq d-1$ *and starting point* $\mathbf{x}_0$*, there is a function* f *in the problem class of* B*-Lipschitz functions over* $\mathbb{R}^d$*, such that any (sub)gradient method has an objective error at least*

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \geq \frac{RB}{2(1+\sqrt{T+1})} \ .$$

Chapter 5

Stochastic Gradient Descent

Sum structured objective functions

Consider sum structured objective functions:

$$f(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}).$$

Here f_i is typically the cost function of the i-th datapoint, taken from a training set of n elements in total.

The SGD algorithm

An iteration of stochastic gradient descent (SGD) is defined as

> sample $i \in [n]$ uniformly at random
> $\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma_t \nabla f_i(\mathbf{x}_t).$

The vector $\mathbf{g}_t := \nabla f_i(\mathbf{x}_t)$ is called a stochastic gradient.

Unbiasedness of a stochastic gradient

Why uniform sampling?

In expectation over the random choice of i, $\mathbf{g}_t$ does coincide with the full gradient of f:

$$\mathbb{E}\left[\mathbf{g}_t \middle| \mathbf{x}_t\right] = \nabla f(\mathbf{x}_t).$$

- $\mathbf{g}_t$ is an unbiased stochastic gradient.

Why SGD?

n times **cheaper**!

Idea: follow the vanilla analysis with $\nabla f(\mathbf{x}_t)$ replaced by $\mathbf{g}_t$...

next week...