

Practical Session on Convex Optimization: Non-Smooth Optimization

Mark Schmidt

INRIA/ENS

September 2011

Motivation: Sparse Regularization

- Consider ℓ_1 -regularized optimization problems,

$$f(x) = g(x) + \lambda \|x\|_1,$$

where g is differentiable.

- The objective is *non-differentiable* when any $x_i = 0$.

Motivation: Sparse Regularization

- Consider ℓ_1 -regularized optimization problems,

$$f(x) = g(x) + \lambda \|x\|_1,$$

where g is differentiable.

- The objective is *non-differentiable* when any $x_i = 0$.
- Similar non-differentiabilities arise for group ℓ_1 -regularization, TV-regularization, nuclear-norm regularization, etc..
- How can we solve non-smooth convex optimization problems?

A vector d is a **subgradient** of f at x if

$$f(y) \geq f(x) + d^T(y - x), \forall y.$$

A vector d is a **subgradient** of f at x if

$$f(y) \geq f(x) + d^T(y - x), \forall y.$$

- The set of all sub-gradients of f at x is called the **sub-differential**, denoted $\partial f(x)$.
- For convex f , $\partial f(x)$ is non-empty, convex, and compact.
- If $f(x) = f_1(x) + f_2(x)$, then $\partial f(x) = \partial f_1(x) + \partial f_2(x)$.
- A convex f is differentiable at x iff sub-gradient is unique.

A vector d is a **subgradient** of f at x if

$$f(y) \geq f(x) + d^T(y - x), \forall y.$$

- The set of all sub-gradients of f at x is called the **sub-differential**, denoted $\partial f(x)$.
- For convex f , $\partial f(x)$ is non-empty, convex, and compact.
- If $f(x) = f_1(x) + f_2(x)$, then $\partial f(x) = \partial f_1(x) + \partial f_2(x)$.
- A convex f is differentiable at x iff sub-gradient is unique.
- Note that $0 \in \partial f(x)$ implies that x is a global minimum.

Sub-Differential of ℓ_1 -Regularization Problem

- The sub-differential of ℓ_1 -regularized optimization problems,

$$f(x) = g(x) + \lambda \|x\|_1,$$

is the set

$$\partial f(x) = \nabla f(x) + \lambda \sum_i \partial |x_i|,$$

- Compute the sub-differential $\partial |x_i|$.

Exercise: Sub-gradient method

- The sub-gradient descent method:

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

for some $d_k \in \partial f(x_k)$.

- For convergence, we require $\sum_i \alpha_k = \infty$ and $\alpha_k \rightarrow 0$.

Exercise: Sub-gradient method

- The sub-gradient descent method:

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

for some $d_k \in \partial f(x_k)$.

- For convergence, we require $\sum_i \alpha_k = \infty$ and $\alpha_k \rightarrow 0$.
- **Computational exercise:** modify the `findMin0.m` and `regLogistic.m` functions to implement a sub-gradient method for ℓ_1 -regularized logistic regression.

Exercise: Sub-gradient method

- The sub-gradient descent method:

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

for some $d_k \in \partial f(x_k)$.

- For convergence, we require $\sum_i \alpha_k = \infty$ and $\alpha_k \rightarrow 0$.
- Computational exercise:** modify the `findMin0.m` and `regLogistic.m` functions to implement a sub-gradient method for ℓ_1 -regularized logistic regression.
- Theoretical exercise:** show that $[f(x_{\min}) - f(x_*)]$ is in $O(1/\sqrt{k})$ with

$$\alpha_k = \|x_0 - x_*\|^2 / G\sqrt{k}$$

provided that for all x we have

$$\|d\| \leq G, \forall d \in \partial f(x).$$

Hints: use definition of x_k in $\|x_k - x_*\|^2$, group together $(x_{k-1} - x_*)$, expand, use definition of sub-gradient to introduce $f(x_k) - f(x_*)$, bound sum of $f(x_{\min}) - f(x_*)$.

Some of the most widely-used alternatives to sub-gradient method:

- Cutting-plane and bundle methods, still have rate $O(1/\sqrt{k})$.
- Smoothing methods, can get rate down to $O(1/k)$.
- **Proximal-gradient** methods, can get rate down to $O(1/k^2)$.
(for problems of the form *smooth* + *non-smooth*).

- The proximal-gradient method addresses problem of the form

$$\min_x f(x) = g(x) + h(x),$$

where g is differentiable but h is a general convex function.

Proximal-Gradient Method

- The proximal-gradient method addresses problem of the form

$$\min_x f(x) = g(x) + h(x),$$

where g is differentiable but h is a general convex function.

- It uses iterations of the form

$$x_{k+1} = \text{prox}[x_k - \alpha_k \nabla g(x_k)],$$

where

$$\text{prox}[x] = \arg \min_y \frac{1}{2} \|x - y\|^2 + \alpha_k h(y).$$

Proximal-Gradient Method

- The proximal-gradient method addresses problem of the form

$$\min_x f(x) = g(x) + h(x),$$

where g is differentiable but h is a general convex function.

- It uses iterations of the form

$$x_{k+1} = \text{prox}[x_k - \alpha_k \nabla g(x_k)],$$

where

$$\text{prox}[x] = \arg \min_y \frac{1}{2} \|x - y\|^2 + \alpha_k h(y).$$

- Gradient and projected-gradient methods are special cases.
- Convergence rate is the same as the gradient method.
- Can do many of the same tricks (i.e. Armijo line-search, polynomial interpolation, Nesterov, Barzilai-Borwein).

Useful Properties of Proximal-Operator

The following are three useful properties of the prox operator:

- Non-expansiveness:

$$\|\text{prox}[x] - \text{prox}[y]\| \leq \|x - y\|.$$

- Solution is fixed point:

$$x_* = \text{prox}[x_* - \alpha_k \nabla g(x_*)].$$

- Relationship to sub-differential:

$$u = \text{prox}(x) \Leftrightarrow (x - u) \in \partial h(u).$$

Exercise: Proximal-Gradient Method

- The proximal-gradient method:

$$x_{k+1} = \text{prox}[x_k - \alpha_k \nabla g(x_k)],$$

where

$$\text{prox}[x] = \arg \min_y \frac{1}{2} \|x - y\|^2 + \alpha_k h(y).$$

Exercise: Proximal-Gradient Method

- The proximal-gradient method:

$$x_{k+1} = \text{prox}[x_k - \alpha_k \nabla g(x_k)],$$

where

$$\text{prox}[x] = \arg \min_y \frac{1}{2} \|x - y\|^2 + \alpha_k h(y).$$

- Computational Exercise:** Modify either the `findMinNesterov.m` or `findMinScaled.m` code from the first session to do proximal-gradient steps for ℓ_1 -regularized logistic regression.

Hints: the proximal operator is $x_i = \text{sign}(x_i) \max\{0, |x_i| - \lambda\alpha_k\}$, and you need to use $g(x)$ for the gradient step but $f(x)$ in the line search.

Exercise: Proximal-Gradient Method

- The proximal-gradient method:

$$x_{k+1} = \text{prox}[x_k - \alpha_k \nabla g(x_k)],$$

where

$$\text{prox}[x] = \arg \min_y \frac{1}{2} \|x - y\|^2 + \alpha_k h(y).$$

- Computational Exercise:** Modify either the `findMinNesterov.m` or `findMinScaled.m` code from the first session to do proximal-gradient steps for ℓ_1 -regularized logistic regression.

Hints: the proximal operator is $x_i = \text{sign}(x_i) \max\{0, |x_i| - \lambda \alpha_k\}$, and you need to use $g(x)$ for the gradient step but $f(x)$ in the line search.

- Theoretical Exercise:** Show that if $\mu I \preceq \nabla^2 g(x) \preceq LI$ for all x , the proximal-gradient method with a step size of $\alpha_k = 1/L$ has a convergence rate of $\|x_k - x_*\| \leq (1 - \mu/L)^k \|x_0 - x_*\|$.

Hints: use the first two properties in $\|x_{k+1} - x_*\|^2$, and that the assumptions on $\nabla^2 g(x)$ imply

$$\langle \nabla g(x) - \nabla g(y), x - y \rangle \geq \frac{1}{L + \mu} \|\nabla g(x) - \nabla g(y)\|^2 + \frac{L\mu}{L + \mu} \|x - y\|^2,$$

and that $\mu I \preceq \nabla^2 g(x)$ implies $-\|\nabla g(x_k) - \nabla g(x_*)\|^2 \leq -\mu^2 \|x_k - x_*\|^2$.

Steepest Descent for Non-Smooth Optimization

- Convex functions have **directional derivatives** everywhere.

Steepest Descent for Non-Smooth Optimization

- Convex functions have **directional derivatives** everywhere.
- The **steepest descent** direction minimizes the directional derivative.
- If f is differentiable, the steepest descent direction is $-\nabla f(x)$.

Steepest Descent for Non-Smooth Optimization

- Convex functions have **directional derivatives** everywhere.
- The **steepest descent** direction minimizes the directional derivative.
- If f is differentiable, the steepest descent direction is $-\nabla f(x)$.
- For general convex functions, the steepest descent direction is

$$-\arg \min_{d \in \partial f(x)} \|d\|.$$

Exercise: Non-Smooth Steepest Descent

- The 'clipped' steepest descent method for ℓ_1 -regularized optimization:

$$x_{k+1} = \mathcal{P}[x_k - \alpha_k d_k],$$

where

$$d_k = \arg \min_{d \in \partial f(x)} \|d\|,$$

and the \mathcal{P} operator sets variables to zero that change sign.

Exercise: Non-Smooth Steepest Descent

- The 'clipped' steepest descent method for ℓ_1 -regularized optimization:

$$x_{k+1} = \mathcal{P}[x_k - \alpha_k d_k],$$

where

$$d_k = \arg \min_{d \in \partial f(x)} \|d\|,$$

and the \mathcal{P} operator sets variables to zero that change sign.

- **Computational Exercise:** Modify either the [findMinNesterov.m](#) or [findMinScaled.m](#) code from the first session to do 'clipped' steepest descent for ℓ_1 -regularized logistic regression.
- **Theoretical Exercise:** Does this converge without having $\alpha_k \rightarrow 0$?

Most of this lecture is based on material from Dimitri Bertsekas' books on optimization and convex analysis, Arkadi Nemirovski's "Efficient Methods in Convex Programming", and Yuri Nesterov's "Introductory Lectures on Convex Optimization".