Modern Regression

Jean-Luc Bouchot

School of Mathematics and Statistics
Beijing Institute of Technology
jlbouchot@bit.edu.cn

Spring 2021

Outline

Definition 2.1

We distinguish various types of optimization:

$$\underset{x \in \mathbb{R}^d}{\operatorname{argmin}} f(x) \qquad \textit{(unconstrained optimization)}$$

$$\underset{x \in \Omega \subset \mathbb{R}^d}{\operatorname{argmin}} f(x) \qquad \textit{(constrained optimization)}$$

$$\underset{x \in \mathbb{R}^d}{\operatorname{argmin}} f(x) + g(x) \qquad \textit{(regularized optimization)}$$

Of course, there is a possibility for a regularized optimization to also be constrained.

Definition 2.2

In the previous definition f is usually called the **data fidelity term** *while Ω is called the* **feasible set**. *The regularization term g is called the* **model fitting term**.

Remark 2.1

There is a very close relationship between constrained optimization/feasible set and regularized problem/model fitting term.
This close relationship will be very clear in the algorithmic developments.

Definition 2.3

A **local numerical optimization** *algorithm is an iterative algorithm where*

$$x^{k+1} = x^k + \alpha_k d_k$$

assuming a starting point x^0 is provided.
The algorithm is characterized by

- *A choice of direction d_k at each iteration.*
- *A choice of step size α_k at each iteration.*

Example 2.1

We may have already seen some iterative local optimization algorithms:

- Gradient descent: assumes the objective function of an unconstrained problem is differentiable and choose the steepest descent direction: $d_k = -\nabla f(x^k)$.
- Newton-like algorithms: assumes a twice differentiable function and picks $d_k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k)$.
- Quasi-Newton type: approximate the (inverse) Hessian, pick $d_k = -B_k \nabla f(x_k)$ where $B_k \approx \nabla^2 f(x_k)^{-1}$ (SR1 and BFGS are great candidates)

Example 2.2

They are various ways of selecting the step size

- Constant step – Works in the convex settings, if you know a lot about your function. It should be avoided in most cases
- α_k satisfies the Goldstein conditions. Roughly speaking, it makes sure that the next step decreases the objective value sufficiently.
- α_k satisfies the (weak/strong) Wolfe conditions. Roughly speaking, it makes sure that we decrease the function sufficiently, and that decrease at the next point is not as big as at the previous.
- Backtracking α_k: go somewhat far from x^k and reduce slightly the step size until enough decrease is noticed.

Definition 2.4 (Globally convergent algorithms)

An algorithm is said to be **globally convergent** *if*

$$\|\nabla f(x^k)\| \to 0$$

as $k \to \infty$.

Example 2.3

Note that globally convergence only means convergence to a stationary point. As a counter example think of

$$x \mapsto x^3.$$

Remark 2.2

Note that the optimization methods presented above are **local** ...

Remark 2.3

In the unconstrained case(feasible set is the whole space $\mathbb{R}^d$) and for differentiable functions f and g, the gradient descent method is *easy*.

Proposition 2.1

Assume you want to solve the following optimization problem:

$$\min_{x \in \Omega} f(x),$$

where f is a differentiable function and Ω is a closed convex set. Then the gradient descent steps need to be projected onto the feasible set at each iterations:

$$x^{k+1} = P_\Omega(x^k - \alpha_k \nabla f(x^k)).$$

Here P_Ω denotes the projection onto the feasible set Ω.
The algorithm described by these updates corresponds to the **projected gradient descent**.

Definition 2.5

Let $\Omega \subset \mathbb{R}^d$ *be a closed convex set. The projection operator onto* Ω *is defined as*

$$x^* = P_\Omega(x) = \operatorname*{argmin}_{y \in \Omega} \|x - y\|_2^2.$$

Example 2.4

Assume $\Omega = B_0^2(R) = \{x \in \mathbb{R}^d : \|x\|_2^2 \leq R^2$. Then the projection is defined as

$$P_\Omega(x) = \begin{cases} R\frac{x}{\|x\|_2} & \text{if } \|x\|_2 > R \\ x & \text{otherwise.} \end{cases}$$

Proposition 2.2

We can revisit the LASSO problem from the point of view of project or projected gradient descent. We want to solve the following problem:

$$\underset{\|x\|_1 \leq R}{\operatorname{argmin}} \|Ax - y\|_2^2.$$

The solution can be obtained by iteratively computing

$$x^{k+1} = P_\Omega \left(x^k - \gamma_k A^T (Ax^k - y) \right).$$

This is known as the **ISTA** *(Iterative Shrinkage Thresholding Algorithm).*

Remark 2.4

The projection onto the ℓ_1 unit ball is done in $\mathcal{O}(d\log(d))$ operations[a] via a soft thresholding.
However, the threshold value isn't clear at the moment.

[a] this can be further reduced to $\mathcal{O}(d)$ but it goes way beyond the scope of this class

Remark 2.5

What happens if we have a non-differentiable and not a projection?

$$\min_{x \in \mathbb{R}^d} f(x) + g(x)$$

where f is differentiable but g is not?

Definition 2.6

Let $g : \mathbb{R}^d \to \mathbb{R}$ be a convex function. We define its proximal operator as

$$\operatorname{prox}_{g,\gamma}(x) := \operatorname*{argmin}_{y \in \mathbb{R}^d} \{ g(y) + \frac{1}{2\gamma} \|y - x\|_2^2 \}.$$

Definition 2.7

Let f be a (convex) differentiable function and g be a convex function. We define the **proximal gradient descent** *as*

$$x^{k+1} = \operatorname{prox}_{g,\gamma}\left(x^k - \gamma \nabla f(x^k)\right).$$

Example 2.5

Let $g(x) := \|x\|_1$. Its proximal operator is precisely the soft thresholding operator.

$$\operatorname{prox}_{g,\gamma}(x) := S_\gamma(x)$$

where the soft thresholding operator is defined componentwise as

$$S_\gamma(t) = \operatorname{sign}(t) \max\{|t| - \lambda, 0\}.$$

This allows us to justify the ISTA once again!