High-dimensional Statistical Methods

Skoltech

Q. Paris[*]

National Research University HSE
Faculty of Computer Science
Moscow, Russia

Chapter 1. High-dimensional regression

Lecture 2

Practice of constrained least squares

[*] email:qparis@hse.ru, teaching material: http://www.qparis-math.com/teaching.

In this lecture, we discuss practical optimization procedures to compute the ℓ_1-constrained LS estimator. In particular, we present the classical projected gradient descent algorithm as well as the more computationally friendly alternative known as conditional gradient descent a.k.a. the Frank-Wolfe algorithm (Frank and Wolfe, 1956). We first present the methods in a general framework and then investigate their use in the context of constrained LS. For more details and proofs, we refer the reader to Bubeck (2015).

1 Projected gradient descent

We present the algorithm for smooth functions. Let $U \subset \mathbf{R}^p$ be an open set and $f : U \to \mathbf{R}$ be a smooth function. Suppose that for a closed convex set $C \subset U$, the function $f : C \to \mathbf{R}$ is convex. To solve the optimization problem

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

the projected gradient descent algorithm proceeds as follows where $\pi : \mathbf{R}^p \to C$ denotes the projection onto the closed convex C.

Algorithm 1 Projected gradient descent for smooth functions

```
procedure PGD($t, (\gamma_s)_{1\le s\le t-1}$)        ▷ $t \geq 2$.
    select $x_1 \in C$
    for $s = 1 : t-1$ do
        $x_{s+1} = \pi\left(x_s - \gamma_s \nabla f(x_s)\right)$
    return $x_t$
```

While the performance of the projected gradient descent can be studied under more general assumptions (sub-differentiable Lipschitz functions for instance), we'll consider next that f satisfies the following smoothness property adapted to the applications we have in mind.

Definition 1.1. *Given a norm $\|.\|$ on $\mathbf{R}^p$, the function $f : U \to \mathbf{R}$ is said L-smooth on U with respect to $\|.\|$ if it is differentiable on U and if its gradient is L- Lipschitz continuous with respect to $\|.\|$, i.e. if*

$$\|\nabla f(x) - \nabla f(y)\|_\star \leq L\|x - y\|,$$

for all x, $y \in U$ where we have denoted $\|g\|_\star = \sup\{g^\top x : \|x\| \leq 1\}$ the dual norm of $\|.\|$. Note in particular that $\|.\|_\star = \|.\|$ if $\|.\|$ is the Euclidean norm and that $\|.\|_\star = \|.\|_\infty$ if $\|.\| = \|.\|_1$.

As described by the next result, gradient descent converges to the minimum of f at the dimension-free rate $O(1/t)$ under the previous smoothness assumption.

Theorem 1.1 (Theorem 3.7 in Bubeck, 2015). *Suppose $f : U \to \mathbf{R}$ is L-smooth on U with respect to the Euclidean norm $\|.\|_2$ and convex on C. Suppose there exists $x^\star \in C$ realizing the minimum of f on C. Then, the output x_t of the projected gradient descent with constant step-size $\gamma = 1/L$ satisfies*

$$f(x_t) - f(x^\star) \leq \frac{3L\|x_1 - x^\star\|_2^2 + f(x_1) - f(x^\star)}{t}.$$

First, observe that this result, while apparently independent of the dimension p, neglects the computational complexity of the actual computation of gradients. For increasing values of p, reaching a prescribed level ε of accuracy supposes indeed a fixed number of steps of order $O(1/\varepsilon)$ but each of these steps requires heavier computations. The computational bottleneck of the method is in fact more often the computation of the projections which, at each step, consist in an optimization problem itself.

2 Conditional gradient descent

Conditional gradient descent, introduced by Frank and Wolfe (1956), is an alternative to projected gradient descent when C is convex and compact. The algorithm is described below.

Algorithm 2 Conditional gradient descent for smooth functions

```
procedure CGD($t, (\gamma_s)_{1\leq s\leq t-1}$)        ▷ $t \geq 2$.
    select $x_1 \in C$
    for $s = 1 : t-1$ do
        $y_s \in \arg\min_{y\in C} \nabla f(x_s)^\top y$
        $x_{s+1} = (1-\gamma_s)x_s + \gamma_s y_s$
    return $x_t$
```

From a computational point of view, the crucial interest of the method, compared to projected gradient descent, is that the projection step is here replaced by a linear optimization in C which can be much simpler. From a more analytical perspective, the advantage of the Frank-Wolfe algorithm is that it adapts to smoothness in an arbitrary norm and exhibits a similar dimension-free bound as projection gradient descent.

Theorem 2.1 (Theorem 3.8 in Bubeck, 2015)**.** *Suppose f is L-smooth on U for some norm $\|.\|$ and convex on the convex compact C. Then, for $t \geq 2$, the output x_t of the conditional gradient descent with step-size $\gamma_s = 2/(1+s)$ satisfies*

$$f(x_t) - f(x^\star) \leq \frac{2LD^2}{1+t},$$

where we have denoted D the diameter of C for the norm $\|.\|$.

Before giving the proof, we need the following simple Lemma.

Lemma 2.1. *Let $U \subset \mathbf{R}^p$ be an open set and $f : U \to \mathbf{R}$ be L-smooth on U for some norm $\|.\|$. Let $C \subset U$ be a convex. Then, for any $x, y \in C$,*

$$|f(x) - f(y) - \nabla f(y)^\top (x-y)| \leq \frac{L}{2}\|x-y\|^2.$$

Proof of Lemma 2.1. Using Taylor's integral representation of $f(x) - f(y)$, the definition of the dual norm $\|.\|_\star$ and the L-smoothness of f, we obtain

$$\begin{aligned}
&|f(x) - f(y) - \nabla f(y)^\top (x-y)| \\
&= |\int_0^1 (\nabla f(y + t(x-y)) - \nabla f(y))^\top (x-y)\,\mathrm{d}t| \\
&\leq \int_0^1 |(\nabla f(y + t(x-y)) - \nabla f(y))^\top (x-y)|\mathrm{d}t \\
&\leq \int_0^1 \|\nabla f(y + t(x-y)) - \nabla f(y)\|_\star \|x-y\|\mathrm{d}t \\
&\leq \int_0^1 Lt\|x-y\|^2\mathrm{d}t \\
&= \frac{L}{2}\|x-y\|^2.
\end{aligned}$$

□

Proof of Theorem 2.1. Recall that a smooth function f is convex on C if and only if, for any $x, y \in C$,

$$\nabla f(x)^\top (y-x) \leq f(y) - f(x).$$

Using Lemma 2.1, the definition of x_{s+1}, the definition of x_s and the convexity of f, it follows that

$$\begin{aligned}
f(x_{s+1}) - f(x_s) &\leq \nabla f(x_s)^\top (x_{s+1} - x_s) + \frac{L}{2}\|x_{s+1} - x_s\|^2 \\
&= \gamma_s \nabla f(x_s)^\top (y_s - x_s) + \frac{L}{2}\gamma_s^2 \|y_s - x_s\|^2 \\
&\leq \gamma_s \nabla f(x_s)^\top (x^\star - x_s) + \frac{L}{2}\gamma_s^2 D^2 \\
&\leq \gamma_s (f(x^\star) - f(x_s)) + \frac{L}{2}\gamma_s^2 D^2.
\end{aligned}$$

Rewriting the inequality in terms of $\delta_s = f(x_s) - f(x^\star)$, we deduce that

$$\delta_{s+1} \leq (1 - \gamma_s)\delta_s + \frac{L}{2}\gamma_s^2 D^2.$$

Choosing $\gamma_s = 2/(1+s)$, and using the fact that $s/(1+s)^2 \leq 2/(2+s)$, it then easily follows by induction that, for all $s \geq 2$, $\delta_s \leq 2LD^2/(1+s)$. □

In addition to being projection-free and independent of the norm in which smoothness is defined, the conditional gradient descent algorithm has an additional extremely attractive property in the context of high-dimensional statistics: the iterates produced by the algorithm are sparse in a sense described in the next result, whose proof stands as almost direct corollary of Theorem 2.1.

Theorem 2.2. *Let $U \subset \mathbf{R}^p$ be an open set and $f : U \to \mathbf{R}$ be L-smooth on U for some norm $\|.\|$. Let $\mathcal{K} \subset U$ be a convex polytope of order $q \geq 2$. Then, for any $t \geq 2$, there exists a point $x_t \in \mathcal{K}$ which is a convex combination of at most t-vertices of $\mathcal{K}$ and such that*

$$f(x_t) - \min_{x \in \mathcal{K}} f(x) \leq \frac{2LD^2}{1+t},$$

where we have denoted D the diameter of $\mathcal{K}$ for the norm $\|.\|$.

Proof of Theorem 2.2. The proof follows from the observation that, if the conditional gradient descent algorithm is initialized at a vertex x_1 of $\mathcal{K}$, then each x_s produced by the algorithm is a convex combination of at most s vertices of $\mathcal{K}$. This observation follows by induction. Indeed,

$$y_s \in \arg\max_{y \in \mathcal{K}} (-\nabla f(x_s)^\top y),$$

can be chosen as a vertex of $\mathcal{K}$ since the function $y \mapsto -\nabla f(x_s)^\top y$ is linear and therefore convex (this last property was proven in Appendix A.1 of the present course). Hence, if x_s is a convex combination of at most s vertices and y_s is a vertex, the definition of x_{s+1} implies the desired conclusion. □

Remark 2.1. *Optimality of the previous result: consider $f(x) = \|x\|_2^2$ on the simplex $\mathcal{K} = \{x \in \mathbf{R}_+^p : \sum_{j=1}^p x_j = 1\}$. Then, show that for any vector $x \in \mathcal{K}$ that is a convex combination of at most t vertices of $\mathcal{K}$ (with $t < p$), we get*

$$f(x) - \min_{\mathcal{K}} f \geq \frac{1}{pt}.$$

Compare to the upper bound provided in the last Theorem.

3 Back to constrained LS

We now discuss the application of conditional gradient descent to the ℓ_1-constrained LS problem. The material of the following paragraph can be found in Bubeck (2015) and is inspired from Lugosi (2010). In this paragraph, we wish to solve the optimization problem

$$\min_{\boldsymbol{\beta} \in \mathcal{K}} f(\boldsymbol{\beta}), \tag{3.1}$$

where

$$f(\boldsymbol{\beta}) = \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{and} \quad \mathcal{K} = \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_1 \leq 1\}.$$

In the sequel, we denote $c(n,p)$ the worst case computational time needed to solve, for some $y \in \mathbf{R}^n$, the linear optimization problem

$$\min_{1 \leq j \leq p} y^\top \mathbf{x}^j,$$

where the vectors $\mathbf{x}^1, \ldots, \mathbf{x}^p \in \mathbf{R}^n$ denote the columns of the design matrix $\mathbf{X}$. Then, we have the following result.

Theorem 3.1. *Let $\varkappa = \max\{\|\mathbf{x}^j\|_2 : 1 \leq j \leq p\}$. For any $\varepsilon > 0$, an ε-optimal solution of (3.1) can be computed in time*

$$O\left(\frac{\varkappa^2 c(n,p)}{\varepsilon} + \frac{\varkappa^2 n}{\varepsilon} + \frac{\varkappa^4}{\varepsilon^2}\right),$$

using conditional gradient descent.

Proof of Theorem 3.1. Let us first observe that the function f is $\varkappa^2$-smooth for the ℓ_1 norm. Indeed, observe that the dual norm of $\|.\|_1$ is $\|.\|_\infty$, that

$\nabla f(\beta) = \mathbf{X}^\top(\mathbf{X}\beta - \mathbf{Y})$ and therefore that, for all $u, v \in \mathbf{R}^p$,

$$\begin{aligned}
\|\nabla f(u) - \nabla f(v)\|_\infty &= \|\mathbf{X}^\top \mathbf{X}(u-v)\|_\infty \\
&= \max_{1\le j\le p} |(\mathbf{x}^j)^\top \sum_{k=1}^p (u_k - v_k)\mathbf{x}^k| \\
&\le \max_{1\le j\le p} \|\mathbf{x}^j\|_2 \| \sum_{k=1}^p (u_k - v_k)\mathbf{x}^k\|_2 \\
&\le \max_{1\le j\le p} \|\mathbf{x}^j\|_2^2 \sum_{k=1}^p |u_k - v_k| \\
&= \varkappa^2 \|u - v\|_1.
\end{aligned}$$

As a result, after t-steps of conditional gradient descent, we get according to Theorem 2.1,

$$f(\boldsymbol{\beta}_t) - f(\boldsymbol{\beta}^\star) \le \frac{8\varkappa^2}{1+t}, \tag{3.2}$$

since here $D = 2$. Next, we study the computational time required for the s-th step of conditional gradient descent. Observe that $\nabla f(\boldsymbol{\beta}_s) = \mathbf{X}^\top \boldsymbol{\alpha}_s$ where $\boldsymbol{\alpha}_s = \mathbf{X}\boldsymbol{\beta}_s - \mathbf{Y}$ and suppose that $\boldsymbol{\alpha}_s$ is already computed. Denote $e_1, \dots, e_p$ the canonical basis vectors in $\mathbf{R}^p$. Then, using arguments already mentioned (indicating that we may restrict the first minimization step of the Frank-Wolfe iteration to vertices) we notice that computing y_s consists in finding the vertex $v_s \in \{\pm e_1, \dots, \pm e_p\}$ which minimizes $v_s^\top \mathbf{X}^\top \boldsymbol{\alpha}_s$. This vertex v_s may be found by minimizing $-\boldsymbol{\alpha}_s{}^\top \mathbf{x}^j$ and $\boldsymbol{\alpha}_s{}^\top \mathbf{x}^j$ in $j \in \{1, \dots, p\}$ and therefore requires a time $2c(n,p)$ by definition. Computing $\boldsymbol{\beta}_{s+1}$ in the second step of the Frank-Wolfe iteration, given $\boldsymbol{\beta}_s$ and v_s, then requires a time proportional to $s+1$ since $\|\boldsymbol{\beta}_s\|_0 \le s$ and $\|v_s\|_0 = 1$. Finally, computing $\boldsymbol{\alpha}_{s+1}$ from β_{s+1} takes a time proportional to n. To sum up, the s-th iteration requires a time of order $O(c(n,p) + n + s)$ and therefore, the time needed for t iterations is of order $O(t(c(n,p) + n) + t^2)$. Finally, to complete the proof, deduce from(3.2) that computing an ε-optimal solution requires at most $t = O(\varkappa^2/\varepsilon)$ iterations. □

Remark 3.1. *A final remark has to be done concerning the context where this computational time is actually low. In the regime where $n \ll p$, an obvious requirement would be for instance that $c(n,p) \le \mathrm{p}(n)$ where $\mathrm{p}(n)$ is polynomial in n. In this case, the upper bound in the last theorem becomes*

$$O\left(\frac{\varkappa^2 \mathrm{p}(n)}{\varepsilon} + \frac{\varkappa^4}{\varepsilon^2}\right).$$

The requirement $c(n,p) \le \mathrm{p}(n)$ is met in the context of some specific structure of the set of columns $\mathbf{x}^j$.

References

S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

G. Lugosi. Comment on: ℓ_1-penalization for mixture regression models. *Test*, 19(2):259–263, 2010.