Statistical machine learning and convex optimization **2016**

Lecture 2 — February 25th

Lecturer: Francis Bach *Scribe: Guillaume Maillard, Nicolas Brosse*

This lecture deals with classical methods for convex optimization. For a convex function, a local minimum is a global minimum and the uniqueness is assured in the case of strict convexity. In the sequel, g is a convex function on $\mathbb{R}^d$. The aim is to find one (or the) minimum $\theta_* \in \mathbb{R}^d$ and the value of the function at this minimum $g(\theta_*)$. Some key additional properties will be assumed for g :

- Lipschitz continuity
$$\forall \theta \in \mathbb{R}^d,\ \|\theta\|_2 \leqslant D \Rightarrow \|g'(\theta)\|_2 \leqslant B.$$
- Smoothness
$$\forall (\theta_1,\ \theta_2) \in \left(\mathbb{R}^d\right)^2,\ \|g'(\theta_1) - g'(\theta_2)\|_2 \leqslant L\|\theta_1 - \theta_2\|_2.$$
- Strong convexity
$$\forall (\theta_1,\ \theta_2) \in \left(\mathbb{R}^d\right)^2, g(\theta_1) \geqslant g(\theta_2) + g'(\theta_2)^\top(\theta_1 - \theta_2) + \tfrac{\mu}{2}\|\theta_1 - \theta_2\|_2^2.$$

We refer to Lecture 1 of this course for additional information on these properties. We point out 2 key references: [1], [2].

2.1 Smooth optimization

If $g : \mathbb{R}^d \to \mathbb{R}$ is a convex L-smooth function, we remind that for all $\theta, \eta \in \mathbb{R}^d$:

$$\|g'(\theta) - g'(\eta)\| \leqslant L\|\theta - \eta\|.$$

Besides, if g is twice differentiable, it is equivalent to:

$$0 \preccurlyeq g''(\theta) \preccurlyeq LI.$$

Proposition 2.1 (Properties of smooth convex functions) *Let $g : \mathbb{R}^d \to \mathbb{R}$ be a convex L-smooth function. Then, we have the following inequalities:*

1. *Quadratic upper-bound:* $0 \leqslant g(\theta) - g(\eta) - g'(\eta)^\top(\theta - \eta) \leqslant \frac{L}{2}\|\theta - \eta\|^2$
2. *Co-coercivity:* $\frac{1}{L}\|g'(\theta) - g'(\eta)\|^2 \leqslant \left[g'(\theta) - g'(\eta)\right]^\top(\theta - \eta)$
3. *Lower bound:* $g(\theta) \geqslant g(\eta) + g'(\eta)^\top(\theta - \eta) + \frac{1}{2L}\|g'(\theta) - g'(\eta)\|^2$
4. *"Distance" to optimum:* $g(\theta) - g(\theta_*) \leqslant g'(\theta)^\top(\theta - \theta_*)$
5. *If g is also μ-strongly convex, then*
$$g(\theta) \leqslant g(\eta) + g'(\eta)^\top(\theta - \eta) + \frac{1}{2\mu}\|g'(\theta) - g'(\eta)\|^2$$

6. If g is also μ-strongly convex, another "distance" to optimum:

$$g(\theta) - g(\theta_*) \leqslant \frac{1}{2\mu}\|g'(\theta)\|^2$$

Proof (1) is a simple application of Taylor expansion with integral remainder.

(3) We define: $h(\theta) = g(\theta) - \theta^\top g'(\eta)$ which is convex with a global minimum at η. Then:

$$h(\eta) \leqslant h(\theta - \frac{1}{L}h'(\theta)) \leqslant h(\theta) + h'(\theta)^\top(-\frac{1}{L}h'(\theta))) + \frac{L}{2}\|-\frac{1}{L}h'(\theta))\|^2 \leqslant h(\theta) - \frac{1}{2L}\|h'(\theta)\|^2$$

Thus $g(\eta) - \eta^\top g'(\eta) \leqslant g(\theta) - \theta^\top g'(\eta) - \frac{1}{2L}\|g'(\theta) - g'(\eta)\|^2$

(2) Apply (3) twice for (η, θ) and (θ, η), and sum to get

$$0 \geqslant [g'(\eta) - g'(\theta)]^\top(\theta - \eta) + \frac{1}{L}\|g'(\theta) - g'(\eta)\|^2$$

(4) is immediate from the definition of convex function.

(5) We define $h(\theta) = g(\theta) - \theta^\top g'(\eta)$ which is convex with a global minimum at η.

$$h(\eta) = \min_\theta h(\theta) \geqslant \min_\zeta h(\theta) + h'(\theta)^\top(\zeta - \theta) + \frac{\mu}{2}\|\zeta - \theta\|^2$$

The min is attained for: $\zeta - \theta = -\frac{1}{\mu}h'(\theta)$ This leads to $h(\eta) \geqslant h(\theta) - \frac{1}{2\mu}\|h'(\theta)\|^2$ Hence,

$$g(\eta) - \eta^\top g'(\eta) \geqslant g(\theta) - \theta^\top g'(\eta) - \frac{1}{2\mu}\|g'(\eta) -' (\theta)\|^2$$

(6) Put $\eta = \theta_*$, where θ_* is a global minimizer of g in (5). ■

Note 2.1.1 *The proofs are simple with second-order derivatives. For (5) and (6), there is no need of smoothness assumption.*

2.1.1 Smooth gradient descent

Proposition 2.2 (Smooth gradient descent) *Let g be a L-smooth convex function, and θ_* be one (or the) minimum of g. For the following algorithm,*

$$\theta_t = \theta_{t-1} - \frac{1}{L}g'(\theta_{t-1})$$

we have the bounds:

*1. if g **is μ-strongly convex**,*

$$g(\theta_t) - g(\theta_*) \leqslant (1 - \mu/L)^t\big[g(\theta_0) - g(\theta_*)\big]$$

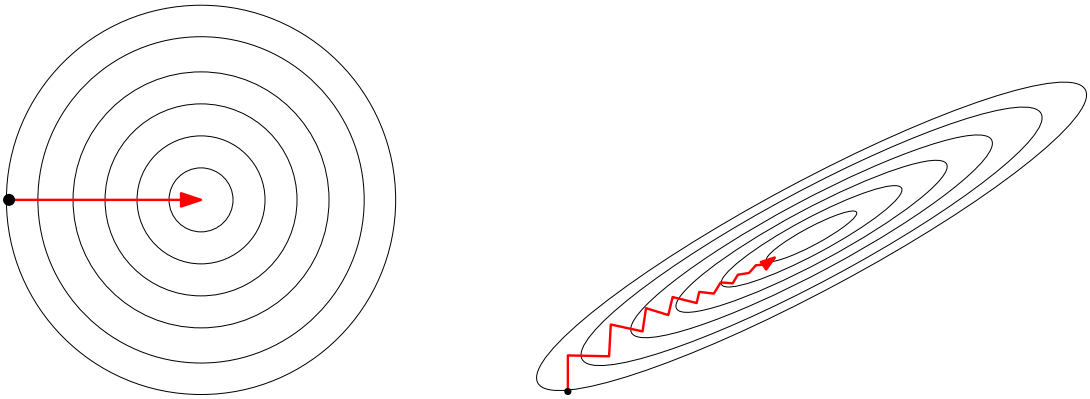

Figure 2.1. Level lines of 2 convex functions. On the left (large μ), best case for gradient descent. On the right (small μ), the algorithm tends to do small steps and is thus slower.

2. if g ***is only*** *L-**smooth**,*

$$g(\theta_t) - g(\theta_*) \leqslant \frac{2L\|\theta_0 - \theta_*\|^2}{t+4}.$$

Note 2.1.2

- *(step-size) In that case, the step-size $(1/L)$ is constant. A line search is also possible (see, e.g.,* `https://en.wikipedia.org/wiki/Line_search`*), which consists of optimizing the step at each iteration in order to find the minimum of g along the direction $g'(\theta_{t-1})$. However, it can be time-consuming and should be carefully used (in particular, no need to be very precise in each line search).*
- *(robustness) For the same algorithm, we obtain different bounds depending on the assumptions that we make on g. It is important to notice that the algorithm is identical in all cases and adapts to the difficulty of the situation.*
- *(lower bounds) It is not the best possible convergence rates after $O(d)$ iterations.*

Simple direct proof for quadratic convex functions In order to illustrate these results, we study the case of quadratic convex functions:

$$g(\theta) = \frac{1}{2}\theta^\top H\theta - c^\top\theta,$$

where H is a symmetric positive semi-definite matrix. We denote by μ and L respectively the smallest and the largest eigenvalues of H. The global optimum is easily derived: $\theta_* = H^{-1}c$ (or $H^\dagger c$ where $H^\dagger$ is the pseudo-inverse of H). Let us compute explicitly the gradient descent iteration:

$$\begin{aligned}
\theta_t &= \theta_{t-1} - \frac{1}{L}(H\theta - c) = \theta_{t-1} - \frac{1}{L}(H\theta - H\theta_*) \\
\theta_t - \theta_* &= (I - \frac{1}{L}H)(\theta_{t-1} - \theta_*) = (I - \frac{1}{L}H)^t(\theta_0 - \theta_*)
\end{aligned}$$

Case 1: We assume strong convexity of g. μ is therefore positive ($\mu > 0$) and the eigenvalues of $(I - \frac{1}{L}H)^t$ are in $[0, (1 - \frac{\mu}{L})^t]$. We obtain an exponential decrease towards 0 and we deduce convergence of iterates:

$$\|\theta_t - \theta_*\|^2 \leqslant (1 - \mu/L)^{2t}\|\theta_0 - \theta_*\|^2.$$

To deal with function values $g(\theta_t) - g(\theta_*)$, we notice that:

$$g(\theta_t) - g(\theta_*) = \frac{1}{2}(\theta - \theta_*)^\top H(\theta - \theta_*),$$

using Taylor expansion at the order 2, or the fact $\theta_* = H^{-1}c$. We thus get:

$$g(\theta_t) - g(\theta_*) \leqslant (1 - \mu/L)^{2t}\big[g(\theta_0) - g(\theta_*)\big].$$

Case 2: g is only L-smooth convex, that is, μ equals 0, and the eigenvalues of $(I - \frac{1}{L}H)^t$ are in $[0, 1]$. We do not have convergence of iterates:

$$\|\theta_t - \theta_*\|^2 \leqslant \|\theta_0 - \theta_*\|^2,$$

but we still have:

$$g(\theta_t) - g(\theta_*) = \frac{1}{2}(\theta - \theta_*)^\top H(\theta - \theta_*),$$

from which we deduce:

$$g(\theta_t) - g(\theta_*) \leqslant \max_{v \in [0,L]} v(1 - v/L)^{2t}\|\theta_0 - \theta_*\|^2$$

by a decomposition along the eigenspaces. For $e \in [0, L]$ and $t \in \mathbb{N}$, we have:

$$v(1 - v/L)^{2t} \leqslant \frac{L}{2t}\left[\frac{2tv}{L}\exp(-\frac{2tv}{L})\right] \leqslant C\frac{L}{2t},$$

and we get (up to a constant):

$$g(\theta_t) - g(\theta_*) \leqslant \frac{L}{t}\|\theta_0 - \theta_*\|^2.$$

Proof (1) One iteration of the algorithm consists of: $\theta_t = \theta_{t-1} - \gamma g'(\theta_{t-1})$ with $\gamma = 1/L$. Let us compute the function value $g(\theta_t)$:

$$\begin{aligned}
g(\theta_t) = \; & g\big[\theta_{t-1} - \gamma g'(\theta_{t-1})\big] \leqslant g(\theta_{t-1}) + g'(\theta_{t-1})^\top\big[-\gamma g'(\theta_{t-1})\big] + \frac{L}{2}\|-\gamma g'(\theta_{t-1})\|^2 \\
= \; & g(\theta_{t-1}) - \gamma(1 - \gamma L/2)\|g'(\theta_{t-1})\|^2 \\
= \; & g(\theta_{t-1}) - \frac{1}{2L}\|g'(\theta_{t-1})\|^2 \text{ if } \gamma = 1/L, \\
\leqslant \; & g(\theta_{t-1}) - \frac{\mu}{L}\big[g(\theta_{t-1}) - g(\theta_*)\big] \text{ using strongly-convex ``distance'' to optimum}
\end{aligned}$$

Thus,

$$g(\theta_t) - g(\theta_*) \leqslant (1 - \mu/L)^t\big[g(\theta_0) - g(\theta_*)\big]$$

We may also get [1][p.70, thm 2.1.15]:

$$\|\theta_t - \theta_*\|^2 \leqslant \big(1 - \frac{2\gamma\mu L}{\mu + L}\big)^t \|\theta_0 - \theta_*\|^2$$

as soon as $\gamma \leqslant \frac{2}{\mu+L}$

(2) One iteration of the algorithm consists of: $\theta_t = \theta_{t-1} - \gamma g'(\theta_{t-1})$ with $\gamma = 1/L$. First, we bound the iterates:

$$\begin{aligned}
\|\theta_t - \theta_*\|^2 &= \|\theta_{t-1} - \theta_* - \gamma g'(\theta_{t-1})\|^2 \\
&= \|\theta_{t-1} - \theta_*\|^2 + \gamma^2 \|g'(\theta_{t-1})\|^2 - 2\gamma(\theta_{t-1} - \theta_*)^\top g'(\theta_{t-1}) \\
&\leqslant \|\theta_{t-1} - \theta_*\|^2 + \gamma^2 \|g'(\theta_{t-1})\|^2 - 2\frac{\gamma}{L}\|g'(\theta_{t-1})\|^2 \text{ using co-coercivity} \\
&= \|\theta_{t-1} - \theta_*\|^2 - \gamma(2/L - \gamma)\|g'(\theta_{t-1})\|^2 \leqslant \|\theta_{t-1} - \theta_*\|^2 \text{ if } \gamma \leqslant 2/L \\
&\leqslant \|\theta_0 - \theta_*\|^2
\end{aligned}$$

Using one equality in the proof of (1), we get:

$$g(\theta_t) \leqslant g(\theta_{t-1}) - \frac{1}{2L}\|g'(\theta_{t-1})\|^2$$

By definition of convexity and Cauchy-Schwarz, we have:

$$g(\theta_{t-1}) - g(\theta_*) \leqslant g'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*) \leqslant \|g'(\theta_{t-1})\| \cdot \|\theta_{t-1} - \theta_*\|$$

And finally, putting things together:

$$g(\theta_t) - g(\theta_*) \leqslant g(\theta_{t-1}) - g(\theta_*) - \frac{1}{2L\|\theta_0 - \theta_*\|^2}\big[g(\theta_{t-1}) - g(\theta_*)\big]^2$$

We then have our "Lyapunov" function, a function decreasing when the number of iterates increases. Let denote by α the quantity:

$$\alpha = \frac{1}{2L\|\theta_0 - \theta_*\|^2},$$

and by Δ_k :

$$\Delta_k = g(\theta_k) - g(\theta_*).$$

We then have:

$$\begin{aligned}
\Delta_k &\leqslant \Delta_{k-1} - \alpha\Delta_{k-1}^2 \text{ with } 0 \leqslant \Delta_k = g(\theta_k) - g(\theta_*) \leqslant \frac{L}{2}\|\theta_k - \theta_*\|^2 \\
\frac{1}{\Delta_{k-1}} &\leqslant \frac{1}{\Delta_k} - \alpha\frac{\Delta_{k-1}}{\Delta_k} \text{ by dividing by } \Delta_k\Delta_{k-1} \\
\frac{1}{\Delta_{k-1}} &\leqslant \frac{1}{\Delta_k} - \alpha \text{ because } (\Delta_k) \text{ is non-increasing} \\
\frac{1}{\Delta_0} &\leqslant \frac{1}{\Delta_t} - \alpha t \text{ by summing from } k = 1 \text{ to } t \\
\Delta_t &\leqslant \frac{\Delta_0}{1 + \alpha t \Delta_0} \text{ by inverting} \\
&\leqslant \frac{2L\|\theta_0 - \theta_*\|^2}{t+4} \text{ since } \Delta_0 \leqslant \frac{L}{2}\|\theta_0 - \theta_*\|^2 \text{ and } \alpha = \frac{1}{2L\|\theta_0 - \theta_*\|^2}
\end{aligned}$$

■

2.1.2 Accelerated gradient methods

A natural question arising from the results of this proposition is : can we do better ? Let us state a lower bound using first-order methods.

Definition 2.3 (First-order method) *An algorithm using first-order method is any iterative algorithm that selects θ_t in $\theta_0 + \text{span}(f'(\theta_0), \dots, f'(\theta_{t-1}))$*

Theorem 2.4 (Lower-bound gradient descent) *For every integer $k \leqslant (d-1)/2$ and every θ_0, there exist convex L-smooth functions with a global minimizer θ_* such that for any first-order method,*

$$g(\theta_t) - g(\theta_*) \geqslant \frac{3}{32} \frac{L\|\theta_0 - \theta_*\|^2}{(t+1)^2}$$

Note 2.1.3

- *$k \leqslant (d-1)/2$ is a strong assumption. It means that the number of iterations should not be higher than half the dimension.*
- *$O(1/t)$ rate for gradient method may not be optimal !*

Proof [sketch] We refer to [1][p.58, section 2.1.2] for the complete proof. We let the reader check the following facts as an exercise:

Define the quadratic function

$$g_t(\theta) = \frac{L}{8}\big[(\theta^1)^2 + \sum_{i=1}^{t-1}(\theta^i - \theta^{i+1})^2 + (\theta^t)^2 - 2\theta^1\big]$$

- Fact 1: g_t is L-smooth
- Fact 2: minimizer supported by first t coordinates (closed form)
- Fact 3: any first-order method starting from zero will be supported in the first k coordinates after iteration k
- Fact 4: the minimum over this support in $\{1, \dots, k\}$ may be computed in closed form

Given iteration k, take $g = g_{2k+1}$ and compute lower-bound on $\frac{g(\theta_k)-g(\theta_*)}{\|\theta_0-\theta_*\|^2}$. ■

We now turn to methods susceptible to reach the lower bound stated in theorem 2.4. They are called accelerated gradient methods [3].

Theorem 2.5 (Accelerated gradient descent) *Let g be a convex function with L-Lipschitz-cont. gradient , and minimum attained at θ_*. For the following algorithm,*

$$\begin{aligned}\theta_t &= \eta_{t-1} - \frac{1}{L} g'(\eta_{t-1}) \\ \eta_t &= \theta_t + \frac{t-1}{t+2}(\theta_t - \theta_{t-1})\end{aligned}$$

we have the bound:

$$g(\theta_t) - g(\theta_*) \leqslant \frac{2L\|\theta_0 - \theta_*\|^2}{(t+1)^2}$$

Note 2.1.4

- *For a ten-line proof [4]*
- *This is not improvable (i.e., the scaling in t cannot be less than $O(1/t^2)$ over all similar problems).*

Theorem 2.6 (Accelerated gradient descent) *Let g be a μ-strongly convex function with L-Lipschitz-cont. gradient and minimum attained at θ_*. For the following algorithm,*

$$\begin{aligned}\theta_t &= \eta_{t-1} - \frac{1}{L} g'(\eta_{t-1}) \\ \eta_t &= \theta_t + \frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}(\theta_t - \theta_{t-1})\end{aligned}$$

we have the bound :

$$g(\theta_t) - f(\theta_*) \leqslant L\|\theta_0 - \theta_*\|^2 (1 - \sqrt{\mu/L})^t$$

Note 2.1.5

- *Ten-line proof [4]*
- *Not improvable*
- *Relationship with conjugate gradient for quadratic functions: a similar bound holds.*
- *We need to know μ and L for implementation of the algorithm ! Often difficult.*

2.1.3 Optimization for sparsity-inducing norms

For more information on the subject, see [5]. We use gradient descent as a **proximal method**, a way of approximating non-smooth functions by differentiable ones. Let us introduce the following minimisation problem:

$$\theta_{t+1} = \arg\min_{\theta \in \mathbb{R}^d} f(\theta_t) + (\theta - \theta_t)^\top \nabla f(\theta_t) + \frac{L}{2}\|\theta - \theta_t\|_2^2$$

Since it is a quadratic function in θ, the problem is equivalent to:

$$\theta_{t+1} = \theta_t - \frac{1}{L}\nabla f(\theta_t)$$

More generally, we want to tackle problems of the form: $\min_{\theta\in\mathbb{R}^d} f(\theta) + \mu\Omega(\theta)$

where $\Omega(\theta)$ is here a sparsity-inducing norm. We seek to minimise the number of non-zero terms in θ. The problem is then formulated as:

$$\theta_{t+1} = \arg\min_{\theta\in\mathbb{R}^d} f(\theta_t) + (\theta-\theta_t)^\top \nabla f(\theta_t) + \mu\Omega(\theta) + \frac{L}{2}\|\theta-\theta_t\|_2^2$$

If $\Omega(\theta) = \|\theta\|_1 \Rightarrow$ **Thresholded gradient descent**

In that case, convergence rates are similar than smooth optimization. For acceleration methods, look at [6, 7]. In the following, we illustrate the method by focusing on the soft-thresholding for the ℓ_1-norm with an example in 1 dimension.

Example: quadratic problem in 1D, i.e.

$$\min_{x\in\mathbb{R}} \frac{1}{2}x^2 - xy + \lambda|x|$$

It is a piecewise quadratic function with a kink at zero, with derivatives at 0+: $g_+ = \lambda - y$ and at 0−: $g_- = -\lambda - y$.

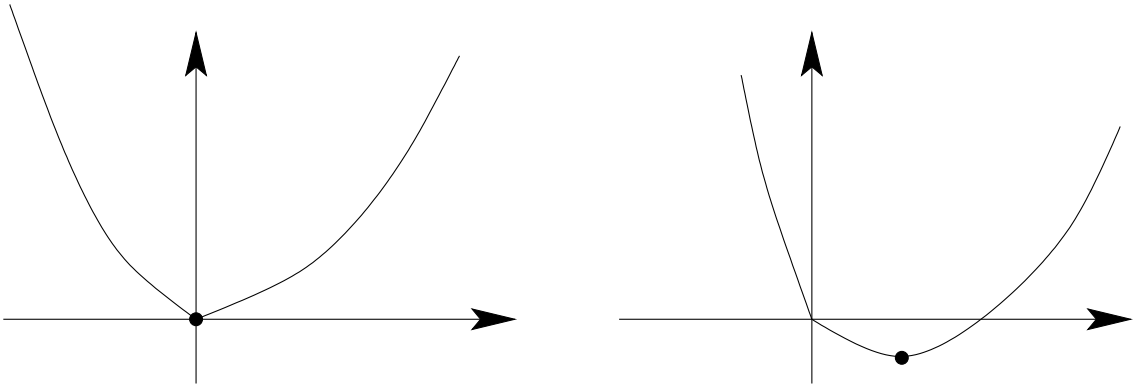

Figure 2.2. Representation of the quadratic function in 1D ($x = 0$ and $x \geqslant 0$)

- $x = 0$ is the solution iff $g_+ \geqslant 0$ and $g_- \leqslant 0$ (i.e., $|y| \leqslant \lambda$)
- $x \geqslant 0$ is the solution iff $g_+ \leqslant 0$ (i.e., $y \geqslant \lambda$) $\Rightarrow x^* = y - \lambda$
- $x \leqslant 0$ is the solution iff $g_- \leqslant 0$ (i.e., $y \leqslant -\lambda$) $\Rightarrow x^* = y + \lambda$

In conclusion, the solution is $\boxed{x^* = \text{sign}(y)(|y| - \lambda)_+}$ = soft thresholding

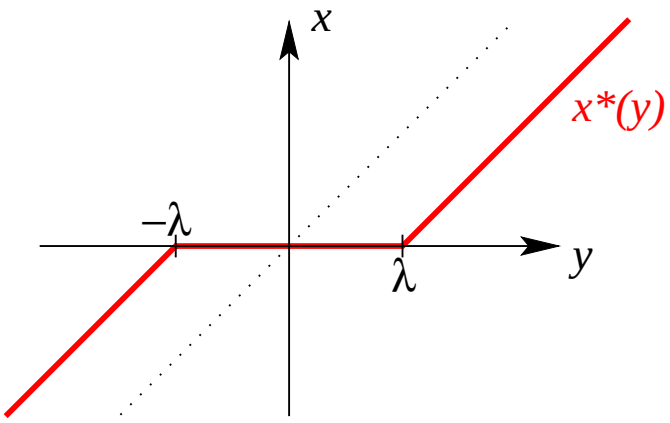

Figure 2.3. Representation of soft thresholding in 1D

2.1.4 Newton method

Given θ_{t-1}, we seek to minimize the second-order Taylor expansion :

$$\tilde{g}(\theta) = g(\theta_{t-1}) + g'(\theta_{t-1})^\top(\theta - \theta_{t-1}) + \frac{1}{2}(\theta - \theta_{t-1})^\top g''(\theta_{t-1})^\top(\theta - \theta_{t-1})$$

It is a quadratic function in θ and we easily derive the minimum:

$$\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$$

The iteration is expensive, and the running-time complexity $O(d^3)$ in general.

We have a property of **quadratic convergence**: if $\|\theta_{t-1} - \theta_*\|$ is small enough, for some constant C, we have

$$(C\|\theta_t - \theta_*\|) = (C\|\theta_{t-1} - \theta_*\|)^2$$

See for example [8].

2.2 Non-smooth optimization

2.2.1 Counter-example: steepest descent for nonsmooth objectives

There is a real difference between the class of smooth problems (with bounded smoothness constant) and the class of all convex problems: methods that converged with a controlled rate in the first setting may now even fail to be consistent. For example, consider the following function:

$$g(\theta_1, \theta_2) = \begin{cases} -5(9\theta_1^2 + 16\theta_2^2)^{\frac{1}{2}} & \theta_1 > |\theta_2| \\ -(9\theta_1 + 16|\theta_2|)^{\frac{1}{2}} & \theta_1 \leq |\theta_2| \end{cases}$$

which is shown in Figure 2.4.
For any value of the stepsize, the fixed stepsize (sub)gradient descent scheme gets stuck at zero, which is clearly not a minimum! New algorithms are needed.

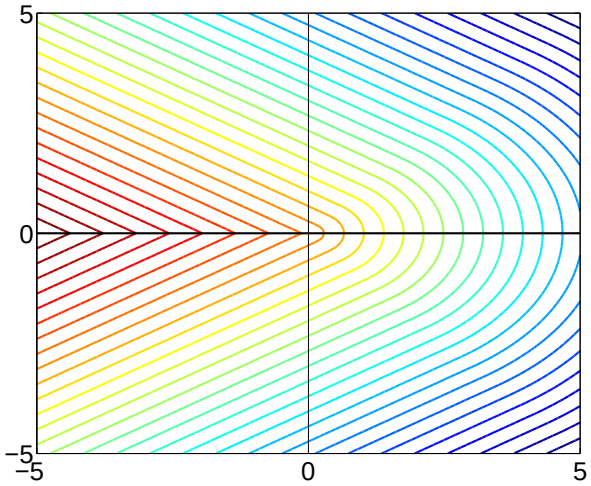

Figure 2.4. Level lines of g. Color indicates function value.

2.2.2 Subgradient method

New setting:
- Convex continuous function g
- Bounded domain $C = \{\theta : \|\theta\|_2 \leq D\}$ although as the proof shows, any compact convex domain with diameter 2D will yield the same result.
- g reaches its minimum on C at θ_*
- g is Lipschitz-continuous on C with constant B

Note 2.2.1 *: In fact, any globally defined, real-valued convex function is locally Lipschitz, so we are dealing with the general convex case.*
Moreover, an upper bound for the Lipschitz constant (for a given value of R) may be calculated with a finite number of function evaluations.

The basic idea is still that of gradient descent, with three important changes:
- Obviously, we will have to replace the gradient with (any possible) subgradient.
- The stepsize will have to decrease with the number of iterates.
- Since even then, iterates may fail to converge, the algorithm does not return the last iterate, but computes an average instead.

Subgradient method: algorithm Let Π_D be the orthogonal projection on C. Then the subgradient algorithm, when run for T iterations, is defined in Algorithm 1.

For an easier proof that shows where the value for the stepsize comes from, we also consider a slightly different algorithm, in which the stepsize is constant over a single run of the algorithm (Algorithm 2). This is known as the finite-horizon setting.

Algorithm 1 minimize g over $C = \{\theta : \|\theta\|_2 \leq D\}$

Require: $\varepsilon > 0$, $B \geq 0$ and $D \geq 0$, $\theta_0 \in C$
Ensure: $g(\hat{\theta}) - g(\theta_*) \leq \varepsilon$
  $T \leftarrow \frac{8D^2B^2}{\varepsilon^2}$
  for $t = 1$ **to** T **do**

$$\theta[t] \leftarrow \Pi_C \left(\theta[t-1] - \frac{\sqrt{2}D}{B\sqrt{t}} g'(\theta[t-1]) \right)$$

  end for
  $\hat{\theta} \leftarrow \frac{1}{T} \sum_{t=1}^{T} \theta[t]$
  return $\hat{\theta}$

Algorithm 2 minimize g over $C = \{\theta : \|\theta\|_2 \leq D\}$

Require: $\varepsilon > 0$, $B \geq 0$ and $D \geq 0$, $\theta_0 \in C$
Ensure: $g(\hat{\theta_c}) - g(\theta_*) \leq \varepsilon$
  $T \leftarrow \frac{4D^2B^2}{\varepsilon^2}$
  for $t = 1$ **to** T **do**

$$\theta[t] \leftarrow \Pi_C \left(\theta[t-1] - \frac{2D}{B\sqrt{T}} g'(\theta[t-1]) \right)$$

  end for
  $\hat{\theta} \leftarrow \frac{1}{T} \sum_{t=1}^{T} \theta[t]$
  return $\hat{\theta}$

The advantage of the first algorithm over the simpler version is that it can be run in an online fashion (often referred to as an "anytime" algorithm), i.e. without knowing in advance the number of iterations or the required precision.

Theorem 2.7

$$g(\hat{\theta}) - g(\theta_*) \leq \frac{2\sqrt{2}DB}{\sqrt{T}}$$

Proof Π_C is 1-Lipschitz since it is a projection. Therefore for any θ

$$\begin{aligned} \|\Pi_C(\theta) - \theta_*\|_2 &= \|\Pi_C(\theta) - \Pi_C(\theta_*)\|_2 \, (\theta_* \in C) \\ &\leq \|\theta - \theta_*\|_2 \end{aligned}$$

Let γ_t denote the stepsize.
Then

$$\begin{aligned} \|\theta_t - \theta_*\|_2^2 &\leq \|\theta_{t-1} - \gamma_t g'(\theta_{t-1}) - \theta_*\|_2^2 \, (\text{by the preceding inequality}) && (2.1) \\ &\leq \|\theta_{t-1} - \theta_*\|_2^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top g'(\theta_{t-1}) + \gamma_t^2 \|g'(\theta_{t-1}\|_2^2 && (2.2) \\ &\leq \|\theta_{t-1} - \theta_*\|_2^2 + B^2\gamma_t^2 - 2\gamma_t(\theta_{t-1} - \theta_*)^\top g'(\theta_{t-1}) (\|g'(\theta)\|_2 \leq B) && (2.3) \\ &\leq \|\theta_{t-1} - \theta_*\|_2^2 + B^2\gamma_t^2 - 2\gamma_t[g(\theta_{t-1}) - g(\theta_*)] (\text{property of subgradients}) && (2.4) \end{aligned}$$

Thus we get:

$$g(\theta_{t-1}) - g(\theta_*) \leq \frac{B^2\gamma_t}{2} + \frac{1}{\gamma_t}[\|\theta_{t-1} - \theta_*\|_2^2 - \|\theta_t - \theta_*\|_2^2].$$

We then average over $1 \leq t \leq T$ Here there is a difference between the two versions of the algorithm:
- If $\gamma_t = \gamma =$ constant then we get a telescoping sum:

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1} g(\theta_t) - g(\theta_*) &\leq \frac{B^2\gamma}{2} + \frac{1}{\gamma T}[\|\theta_0 - \theta_*\|_2^2 - \|\theta_T - \theta_*\|_2^2] \\ &\leq \frac{B^2\gamma}{2} + \frac{1}{\gamma T}\|\theta_0 - \theta_*\|_2^2 \\ &\leq \frac{B^2\gamma}{2} + \frac{D^2}{\gamma T}. \end{aligned}$$

We can then optimize the upper bound in the parameter γ: $\gamma = \frac{2D}{B\sqrt{T}}$ gives $\frac{2DB}{\sqrt{T}}$ which suggests the right dependency for the 'online' version.

- In the variable γ case we use summation by parts:

$$\begin{aligned}
\sum_{t=1}^{T}[g(\theta_{t-1}) - g(\theta_*)] &\leq \frac{B^2}{2}\sum_{t=1}^{T}\gamma_t + \sum_{t=1}^{T}\frac{1}{\gamma_t}[\|\theta_{t-1} - \theta_*\|_2^2 - \|\theta_t - \theta_*\|_2^2] \\
&\leq \frac{B^2}{2}\sum_{t=1}^{T}\gamma_t + \sum_{t=1}^{T-1}\|\theta_t - \theta_*\|_2^2[\frac{1}{2\gamma_{t+1}} - \frac{1}{2\gamma_t}] + \frac{\|\theta_0 - \theta_*\|_2^2}{2\gamma_1} - \frac{\|\theta_T - \theta_*\|_2^2}{2\gamma_T} \\
&\leq \frac{B^2}{2}\sum_{t=1}^{T}\gamma_t + \sum_{t=1}^{T-1}4D^2[\frac{1}{2\gamma_{t+1}} - \frac{1}{2\gamma_t}] + \frac{4D^2}{2\gamma_1} \\
&= \frac{B^2}{2}\sum_{t=1}^{T}\gamma_t + \frac{4D^2}{2\gamma_T}.
\end{aligned}$$

Now we set $\gamma_t = \frac{xD}{B\sqrt{t}}$ to get:

$$\begin{aligned}
\sum_{t=1}^{T}[g(\theta_{t-1}) - g(\theta_*)] &\leq \frac{B^2}{2}\frac{xD}{B}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} + \frac{4D^2}{2}\frac{B\sqrt{T}}{xD} \\
&= \frac{xBD}{2}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} + \frac{2BD\sqrt{T}}{x} \\
&\leq \frac{xBD}{2}\int_0^T \frac{1}{\sqrt{t}}dt + \frac{2BD\sqrt{T}}{x} \\
&\leq xBD\sqrt{T} + \frac{2BD\sqrt{T}}{x}(\text{because } \frac{d}{dt}\sqrt{t} = \frac{1}{2\sqrt{t}}) \\
&= (x + \frac{2}{x})BD\sqrt{T}.
\end{aligned}$$

which is optimized when the two terms are equal, that is $x = \sqrt{2}$, which gives:

$$\frac{1}{T}\sum_{t=1}^{T} g(\theta_{t-1}) - g(\theta_*) \leq \frac{2\sqrt{2}BD}{\sqrt{T}}.$$

Finally, the convexity inequality:

$$g\left(\frac{1}{T}\sum_{t=0}^{T-1}\theta_t\right) \leq \frac{1}{T}\sum_{t=0}^{T-1} g(\theta_t)$$

gives the desired result in both cases.

■

Subgradient descent with strong convexity When the non-smooth convex function is strongly convex with known parameter μ, it is possible to improve the rate of convergence

Algorithm 3 minimize g over $\{x : \|x\|_2 \leq D\}$

Require: $\mu > 0$, $D \geq 0$, $\theta_0 : \|\theta_0\|_2 \leq D$

Ensure: $\hat{\theta}_t = \frac{2}{(t+1)(t+2)} \sum_0^t (u+1)\theta_u$ where a subscript i indicates value of the variable at iteration i.

$\theta \leftarrow \theta_0$

$\hat{\theta} \leftarrow \theta_0$

for $t \geq 1$ **do**

$\theta \leftarrow \Pi_D \left(\theta - \frac{2}{\mu(t+1)} g'(\theta)\right)$

$\hat{\theta} \leftarrow \frac{t}{t+2}\hat{\theta} + \frac{2}{t+2}\theta$

end for

return $\hat{\theta}$

to $O(\frac{1}{\mu t})$. The algorithm and the proof of its convergence properties are similar in spirit to the previous one. Since here the algorithm may be used in cases where B is unknown we present it as an iterative scheme (Algorithm 3).

Theorem 2.8 *Let g be B-Lipschitz and μ-strongly convex over the set $\{\theta : \|\theta\|_2 \leq D\}$. Then for $\hat{\theta}$ defined by the above algorithm,*

$$g(\hat{\theta}_t) - g(\theta_*) \leq \frac{2B^2}{\mu(t+2)}.$$

Note 2.2.2 *: Somewhat surprisingly, D does not appear explicitly in the upper bound. This is because the conditions that g be B-Lipschitz and μ-strongly convex already impose a limit on the size of g 's domain of definition.*

Proof The proof uses the same ideas as for the general case: the beginning is the same. At (2.4), the assumption of strong-convexity permits a better bound:

$$\|\theta_t - \theta_*\|_2^2 \leq \|\theta_{t-1} - \theta_*\|_2^2 + B^2\gamma_t^2 - 2\gamma_t[g(\theta_{t-1}) - g(\theta_*)] + \frac{\mu}{2}\|\theta_{t-1} - \theta_*\|_2,$$

which leads to:

$$\begin{aligned}
2\gamma_t[g(\theta_{t-1}) - g(\theta_*)] &\leq (1 - \gamma_t\mu)\|\theta_{t-1} - \theta_*\|_2^2 + B^2\gamma_t^2 - \|\theta_t - \theta_*\|_2^2 \\
g(\theta_{t-1}) - g(\theta_*) &\leq \frac{1}{2}(\frac{1}{\gamma_t} - \mu)\|\theta_{t-1} - \theta_*\|_2^2 + \frac{B^2\gamma_t}{2} - \frac{1}{2\gamma_t}\|\theta_t - \theta_*\|_2^2 \\
&\leq \frac{B^2}{\mu(t+1)} + \frac{\mu}{2}[\frac{t-1}{2}]\|\theta_{t-1} - \theta_*\|_2^2 - \frac{\mu}{2}[\frac{t+1}{2}]\|\theta_t - \theta_*\|_2^2 \text{ (here } \gamma_t = \frac{2}{\mu(t+1)}).
\end{aligned}$$

Here as before the idea is to average the iterates, taking advantage of convexity on the left hand side and of telescoping terms on the right:

$$\begin{aligned} g(\hat{\theta}_t) - g(\theta_*) &\leq \frac{2}{(t+1)(t+2)} \sum_{u=1}^{t+1} u[g(\theta_{u-1}) - g(\theta_*)] \\ &\leq \frac{2}{(t+1)(t+2)} \left\{ \frac{B^2}{\mu} \sum_{u=1}^{t+1} \frac{u}{u+1} + \frac{\mu}{4} \sum_{u=1}^{t+1} [u(u-1) \|\theta_{u-1} - \theta_*\|_2^2 - (u+1)u \|\theta_u - \theta_*\|_2^2] \right\} \\ &\leq \frac{2}{(t+1)(t+2)} \left\{ \frac{B^2(t+1)}{\mu} + \frac{\mu}{4}[0 - (t+2)(t+1) \|\theta_{t+1} - \theta_*\|_2^2] \right\} \\ &\leq \frac{2B^2}{\mu(t+2)}. \end{aligned}$$

■

2.2.3 Ellipsoid method

Idea The basic idea of the ellipsoid method is that of dichotomy: by recursively splitting a set along hyperplanes given by subgradients (so that all minimizers are one side of the hyperplane), it is hoped that the diameter of the set might decrease to zero exponentially fast, as in the one dimensional dichotomy method.

To get a good rate of convergence, a natural idea is to split the set at its geometric center. However, calculating the geometric center of a set is in general a difficult problem.
This is where knowing few geometric properties of ellipsoids is useful:

Definition 2.9 *Ellipsoid:*
Given a point $\bar{x} \in \mathcal{R}^d$ and a symmetric positive definite matrix P, the ellipsoid $E(\bar{x}, P)$ is the set:

$$\{x \in \mathcal{R}^d \mid (x - \bar{x})^\top P^{-1}(x - \bar{x}) \leq 1\}$$

The point $\bar{x}$ is the geometric center of $E(\bar{x}, P)$ and the matrix P describes the half-axes lengths (eigenvalues) and directions (eigenvectors).

Proposition 2.10 *Given an ellispoid $E(\bar{x}, P)$ and a hyperplane $H = \{g^\top(x - \bar{x}) \geqslant 0\}$ containing $\bar{x}$, there is a minimum volume ellipsoid $E_+(\bar{x}_+, P_+)$ containing $\{x \in E(\bar{x}, P) | g^\top(x - \bar{x}) \geqslant 0\}$,*
with

$$\bar{x}_+ = \bar{x} - \frac{1}{d+1} \frac{Pg}{\|g\|_{2,P}} \tag{2.5}$$

$$P_+ = \frac{d^2}{d^2-1}(P - \frac{2}{d+1} \frac{Pgg^\top P}{\|g\|_{2,P}^2}) \tag{2.6}$$

where $\|g\|_{2,P}$ is the intrinsic P-norm of g, $\sqrt{g^\top P g}$.
Its volume decreases by a constant factor relative to the original ellipsoid:

$$\mathrm{vol}_d(E_+(\bar{x}_+, P_+)) \leq e^{-\frac{1}{2d}} \mathrm{vol}_d(E(x, P))$$

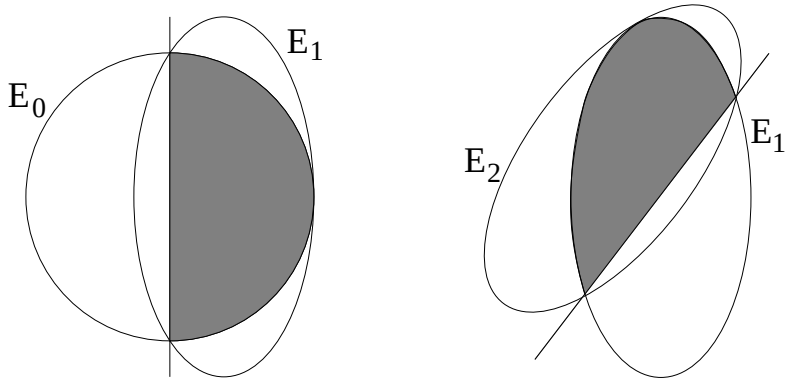

Figure 2.5. 2 cuts and minimal ellipsoids

This suggests the iterative scheme described in Algorithm 4.

Algorithm 4 minimize g over $\{x : \|x\|_2 \leq D\}$

Require: $D \geq 0$, $x_0 : \|x_0\|_2 \leq D$
  $\bar{x}[0] \leftarrow x_0$
  $P \leftarrow D^2 I_d$
  $h \leftarrow 0$
  for $t \geq 1$ **do**
    $h \leftarrow g'(\bar{x}[t-1])$
    $\bar{x}[t] \leftarrow \bar{x}[t-1] - \frac{1}{d+1} \frac{Ph}{\|h\|_{2,P}}$
    $P \leftarrow \frac{d^2}{d^2-1}(P - \frac{2}{d+1} \frac{Phh^\top P}{\|h\|^2_{2,P}})$
  end for
  $t^* \leftarrow \mathrm{argmin}_{0 \leq t \leq stoptime}\, g(\bar{x}[t])$
  return $\bar{x}_{t^*}$

Proposition 2.10 is not quite enough to conclude to a convergence rate, because even if the volume decreases exponentially, it seems hard to ensure that the diameter does the same, as the ellipsoid may be very excentric. However, exponential decrease remains true for approximate minimization. The following proof is taken from Nesterov's book [1].

Definition 2.11 *The following quantity will be useful:*

$$v_g(x) = \frac{1}{\|g'(x)\|_2} (g'(x))^\top (x - x_*)$$

Lemma 2.12 *If g is B-Lipschitz, then*

$$g(x) - g(x_*) \leq B v_g(x)$$

More explicitly,

$$g(x) - g(x_*) \leq \frac{B}{\|g'(x)\|_2} (g'(x))^\top (x - x_*)$$

Proof Let $y = x_* + v_g(x) \frac{g'(x)}{\|g'(x)\|_2}$
Clearly $\|y - x_*\|_2 \leq v_g(x)$
By convexity,

$$g(y) \geq g(x) + (g'(x))^\top (y - x)$$

But by definition,

$$(g'(x))^\top (x - y) = (g'(x))^\top (x - x_*) - v_g(x) \|g'(x)\|_2 = 0$$

In the end:

$$g(x) - g(x_*) \leq g(y) - g(x_*) \leq B \|y - x_*\|_2 \leq B v_g(x)$$

■

The main point of introducing this new quantity is that it can be bounded by the volume:

Proposition 2.13 *Let Q be a convex set in $\mathcal{R}^d$ with finite diameter D and nonempty interior. Let $x_1, ..., x_n \in \mathcal{R}^d$ be points and $(E_i)_{i=1,..,n}$ be sets such that*

$$\{x \in Q | \forall i \in \{1, \ldots, n\}, (g'(x_i))^\top (x_i - x) \geq 0\} \subset E_i$$

Assume that $x_* \in E_0 = Q$. Then:

$$\min_{1 \leq i \leq n} v_g(x_i) \leq D \left[\frac{\mathrm{vol}_d E_i}{\mathrm{vol}_d Q} \right]^{\frac{1}{d}}$$

Proof we introduce some notation:
let $v_n^* = \min_{1 \leq i \leq n} v_g(x_i)$
and $\alpha = \frac{v_n^*}{D}$.

$$(1 - \alpha) x_* + \alpha Q \subset (1 - \alpha) x_* + \alpha B(x_*, D) = B(x_*, v_n^*)$$

By convexity, we also have that

$$(1 - \alpha) x_* + \alpha Q = [(1 - \alpha) x_* + \alpha Q] \cap Q \subset B(x_*, v_n^*) \cap Q$$

By definition of v_n^*,

$$\begin{aligned}
\|x - x_*\|_2 \leq v_n^* &\Rightarrow \forall i \in \{1, \ldots, n\}, \\
\frac{1}{\|g'(x_i\|_2} (g'(x_i))^\top (x_i - x) &= \frac{1}{\|g'(x_i\|_2} (g'(x_i))^\top (x_i - x_*) + \frac{1}{\|g'(x_i\|_2} (g'(x_i))^\top (x_* - x) \\
&\geq v_n^* - \|x - x_*\|_2 \\
&\geq 0 \\
&\Rightarrow x \in E_i
\end{aligned}$$

Therefore:

$$\alpha^d \mathrm{vol}_d(Q) \leq \mathrm{vol}_d((1-\alpha)x_* + \alpha Q) \leq \mathrm{vol}_d(E_i).$$

■

Combining all three inequalities, we obtain the convergence rate of the ellipsoid method:

Theorem 2.14 *Let g be convex and B-Lipschitz on B(0,D).*
Let $\bar{x}_{n^*}$ be the value returned by Algorithm 4 when stopped at iteration n. Then

$$g(\bar{x}_{n^*}) - g(x_*) \leq 2BDe^{-\frac{n}{2d^2}}.$$

2.2.4 Application to Machine Learning

After having described many different methods, we now review their applicability to machine learning.

Setting As seen in Lecture 1, in machine learning, more specifically in empirical risk minimization with linear predictors, the objective function is of the form:

$$\hat{f}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \Phi(x_i)^\top \theta)$$

Assuming bounded features:

$$\|\Phi(x)\|_2 \leq R$$

and that the loss is G-Lipschitz on $\{\theta | \, \|\theta\|_2 \leq RD\}$, we have shown that $\hat{f}$ is GR-Lipschitz on $\{\theta | \, \|\theta\|_2 \leq D\}$.

Which methods can be used? Reasonable hypothesis in this setting

- Smoothness:
$\hat{f}$ may or may not be smooth, depending on ℓ (notably the hinge loss is unsmooth, but the ridge loss is smooth). If smooth, the constant is essentially the same as for ℓ, times R, by linearity of differenciation.

- Strong convexity:
In typical high-dimensional settings, $\hat{f}$ will rarely be strongly convex for a useful value of μ (in particular it cannot be if $n < d$).

- Dimensional complexity:
High-dimensionality makes the ellipsoid method useless, despite its geometric convergence. This is because the rate is $O(e^{-\frac{n}{2d^2}})$ so $O(d^2)$ iterates are needed. Each iterate requires multiplying a vector by a matrix, so $O(d^2)$ cost is encountered at each iteration, for a total, prohibitive dependency of $O(d^4)$ in the dimension. By contrast, gradient and subgradient methods converge at a rate independent of the dimension. The only dimension-dependent cost is due to arithmetic operations, and is linear in d. Thus, a gradient or subgradient method should be used, if possible one that does not need μ.

The case of subgradient descent : choosing the number of steps
In the worst, non-smooth case, the convergence rate is $O(\frac{1}{\sqrt{t}})$. This is not a problem since we must remember that there is a statistical error: It is $f(\theta) = \mathbb{E}[\hat{f}(\theta)]$ and not $\hat{f}(\theta)$ that we wish to minimize and we know from the previous lesson that with probability greater than δ:

$$\min_{\theta\in\Theta} \hat{f}(\theta) - \min_{\theta\in\Theta} f(\theta) \leq \max_{\theta\in\Theta} |\hat{f}(\theta) - f(\theta)| \leq \frac{GRD}{\sqrt{n}} \left[2 + \sqrt{2\log(\frac{2}{\delta})}\right]$$

Therefore, with $t = n$ iterations of subgradient descent
we get an optimization error of same order as the statistical error:

$$\hat{f}(\hat{\theta}) - \min_{\eta\in\Theta} \hat{f}(\eta) \leq \frac{2GRD}{\sqrt{n}}.$$

There is not much to gain by iterating for longer (only a factor of 2 at best). The overall complexity is $O(n^2 d)$: it depends only linearly on the dimension.

Bibliography

[1] Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Publishers, 2004.

[2] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015. URL `http://dx.doi.org/10.1561/2200000050`.

[3] Y. Nesterov. A method for solving a convex programming problem with rate of convergence $O(1/k^2)$. *Soviet Math. Doklady*, 269(3):543–547, 1983.

[4] M. Schmidt, N. Le Roux, and F. Bach. Convergence Rates of Inexact Proximal-Gradient Methods for Convex Optimization. Advances in Neural Information Processing Systems (NIPS), 2011.

[5] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[6] Y. Nesterov. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Tech. Rep*, 76, 2007.

[7] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.