Optimization for Data Science

Cristian Perez Jensen

July 7, 2024

Note that these are not the official lecture notes of the taught course, but only notes written by me. As such, it might contain mistakes. The source code can be found at `github.com/cristianpjensen/eth-cs-notes`. If you find a mistake, please create an issue or open a pull request.

Contents

List of symbols

$\doteq$	Equality by definition
$\mathbb{R}$	Set of real numbers
$f : A \to B$	Function f that maps elements of set A to elements of set B
$\boldsymbol{v} \in \mathbb{R}^n$	n-dimensional vector
$\boldsymbol{M} \in \mathbb{R}^{m \times n}$	$m \times n$ matrix
$\boldsymbol{M}^\top$	Transpose of matrix $\boldsymbol{M}$
$\boldsymbol{M}^{-1}$	Inverse of matrix $\boldsymbol{M}$
$\det(\boldsymbol{M})$	Determinant of $\boldsymbol{M}$
$\frac{\mathrm{d}}{\mathrm{d}x} f(x)$	Ordinary derivative of $f(x)$ w.r.t. x at point $x \in \mathbb{R}$
$\frac{\partial}{\partial x} f(\boldsymbol{x})$	Partial derivative of $f(\boldsymbol{x})$ w.r.t. x at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\nabla} f(\boldsymbol{x}) \in \mathbb{R}^n$	Gradient of $f : \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{J} f(\boldsymbol{x}) \in \mathbb{R}^{n \times m}$	Jacobian of $f : \mathbb{R}^n \to \mathbb{R}^m$ at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\nabla}^2 f(\boldsymbol{x}) \in \mathbb{R}^{n \times n}$	Hessian of $f : \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x} \in \mathbb{R}^n$

1 Risk minimization

1.1 Algorithms in data science

In classical algorithm theory, an optimization problem solves a well-defined problem. For example, Kruskal's algorithm computes the minimum spanning tree of a graph. In data science, it is not as well-defined. The starting point is a learning problem, and the optimization typically happens on training data. However, even a perfect result may fail to solve the learning problem, which is a failure of the model in which the optimization algorithm was applied, rather than the optimization algorithm itself.

1.2 Empirical and expected risk

Typically, we have a data source $\mathcal{X}$, equipped with an unknown probability distribution. However, we do have access to independent samples $X_1, \dots, X_n \sim \mathcal{X}$. The goal is to "explain" $\mathcal{X}$ through these samples. More specifically, we have a class $\mathcal{H}$ of hypotheses (possible explanations of $\mathcal{X}$). The goal is then to select the hypothesis $H \in \mathcal{H}$ that best explains $\mathcal{X}$, which we measure by a *risk* (or loss) function $\ell : \mathcal{H} \times \mathcal{X} \to \mathbb{R}$.

The expected risk can be computed by

$$\ell(H) \doteq \mathbb{E}_{\mathcal{X}}[\ell(H, X)].$$

The best explanation is the *expected risk minimizer*,

$$H^\star = \operatorname*{argmin}_{H \in \mathcal{H}} \ell(H).$$

However, since we do not have access to the distribution over $\mathcal{X}$, we cannot compute $\ell(H)$ or $H^\star$. Thus, we need to compute the *probably approximately correct* (PAC) hypothesis. This means that given $\delta, \epsilon > 0$, we want to produce, with probability $1 - \delta$, a hypothesis $\tilde{H} \in \mathcal{H}$ such that

$$\ell(\tilde{H}) \leq \inf_{H \in \mathcal{H}} \ell(H) + \epsilon,$$

meaning that with high probability, we approximately solve the expected risk minimization problem.

However, we can still not compute this, thus we must use our training data to compute the *empirical risk*,

$$\ell_n(H) = \frac{1}{n} \sum_{i=1}^{n} \ell(H, X_i).$$

This is a random variable, because it depends on the training data, which are all random variables, distributed according to the probability distribution of $\mathcal{X}$.

Lemma 1 (Weak law of large numbers)**.** Let $H \in \mathcal{H}$ be a hypothesis. For any $\delta, \epsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that for $n \geq n_0$,

$$|\ell_n(H) - \ell(H)| \leq \epsilon,$$

with probability at least $1 - \delta$.

Given $n \in \mathbb{N}$ and training data $X_1, \ldots, X_n$ from $\mathcal{X}$, we want to produce a hypothesis $\tilde{H}_n$ such that

$$\ell_n(\tilde{H}_n) \leq \inf_{H \in \mathcal{H}} \ell_n(H) + \epsilon.$$

In an ideal world, $\tilde{H}_n$ is also almost the best explanation for the data source $\mathcal{X}$,

$$\ell(\tilde{H}_n) \leq \inf_{H \in \mathcal{H}} \ell(H) + \epsilon.$$

Remark. Note that the weak law of large numbers can only be applied to a *fixed* hypothesis, but not to the data-dependent hypothesis $\tilde{H}_n$. Thus, we are not always in an ideal world scenario.

A sufficient condition for an ideal world scenario is that the weak law of large numbers uniformly holds for all hypotheses with high probability. This leads us to the following theorem.

Theorem 2. Assume that for any $\delta, \epsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that for $n \geq n_0$,

$$\sup_{H \in \mathcal{H}} |\ell_n(H) - \ell(H)| \leq \epsilon,$$

with probability at least $1 - \delta$. Then, for $n \geq n_0$, an approximate empirical risk minimizer $\tilde{H}_n$ is PAC for expected risk minimization, meaning that it satisfies

$$\ell(\tilde{H}_n) \leq \inf_{H \in \mathcal{H}} \ell(H) + 3\epsilon,$$

with probability at least $1 - \delta$.

Proof.

$$\begin{aligned}
\ell(\tilde{H}_n) &\leq \ell_n(\tilde{H}_n) + \epsilon && \text{Follows from } \sup_{H \in \mathcal{H}} |\ell_n(H) - \ell(H)| \leq \epsilon. \\
&\leq \inf_{H \in \mathcal{H}} \ell_n(H) + 2\epsilon && \tilde{H}_n \text{ is an almost best explanation of the training data.} \\
&\leq \inf_{H \in \mathcal{H}} \ell(H) + 3\epsilon, && \text{Follows from } \sup_{H \in \mathcal{H}} |\ell_n(H) - \ell(H)| \leq \epsilon.
\end{aligned}$$

with probability at least $1 - \delta$. ■

It turns out that the assumption made by Theorem 2 holds in many relevant cases, but it is not always true.

In this course, we will not learn how to pick the theory ($\mathcal{H}$ and ℓ), but rather how to solve the optimization problems that arise in empirical risk minimization after the theory has been chosen.

1.3 *The map of learning*

The map of learning can be seen in Figure 1. It plots the empirical risk ($\ell_n(H_n)$, training data) against the expected risk ($\ell(H_n)$, estimated by a validation set). We can only be in the area denoted by "empirical risk minimization", because

$$\begin{aligned}\ell_n(\tilde{H}_n) &\leq \inf_{H\in\mathcal{H}} \ell_n + \epsilon \\ &\leq \ell_n(\tilde{H}) + \epsilon \\ &\leq \ell(\tilde{H}) + 2\epsilon \\ &\leq \ell(\tilde{H}_n) + e\epsilon.\end{aligned}$$

A model is overfit when we have low empirical risk, while having high expected risk. This means that the explanation quality on the data source is much worse than on the training data. The main reason for this is that the theory ($\mathcal{H}$ and ℓ) is so complex that it can explain almost anything.

A model is underfit when we have high empirical risk, while having high expected risk. This means that we neither explain the training data nor the data source. The main reason for this is that the theory is too simple to capture the nature of the data.

The model is learning when we have low empirical risk and low expected risk. This means that the training was successful. Generalization occurs when the expected risk is close to the empirical risk. Note that this does not mean that the explanation is good, since any "blind explanation" will generalize well due to the weak law of large numbers. Ideally, we want generalization and learning.

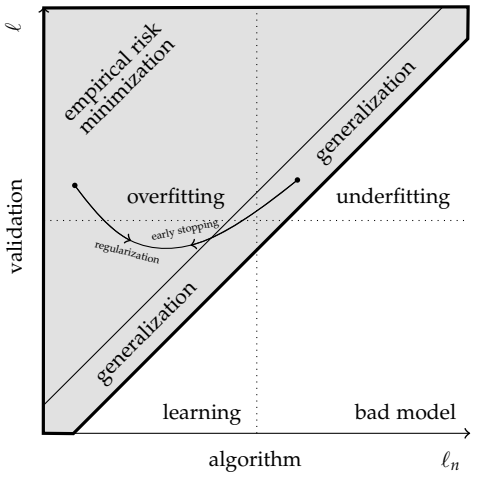


Figure 1. The map of learning.

2 *Theory of convex functions*

2.1 *Mathematical background*

Theorem 3 (Cauchy-Schwarz inequality)**.** Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$, then

$$|\boldsymbol{u}^\top \boldsymbol{v}| \leq \|\boldsymbol{u}\|\|\boldsymbol{v}\|.$$

For non-zero vectors, this is equivalent to

$$-1 \leq \frac{\boldsymbol{u}^\top \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|} \leq 1.$$

Definition 4 (Spectral norm)**.** Let $\boldsymbol{A}$ be an $m \times d$ matrix, then

$$\|\boldsymbol{A}\| \doteq \max_{\boldsymbol{v} \in \mathbb{R}^d, \boldsymbol{v} \neq \boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{v}\|}{\|\boldsymbol{v}\|} = \max_{\|\boldsymbol{v}\|=1} \|\boldsymbol{A}\boldsymbol{v}\|.$$

Intuitively, this means that $\|\boldsymbol{A}\|$ is the largest factor by which a vector can be stretched in length under the mapping $\boldsymbol{v} \mapsto \boldsymbol{A}\boldsymbol{v}$.

Theorem 5 (Mean value theorem)**.** Let $a < b$ be real numbers, and let $h : [a, b] \to \mathbb{R}$ be a continuous function that is differentiable on (a, b). Then, there exists $c \in (a, b)$ such that

$$h'(c) = \frac{h(b) - h(a)}{b - a}.$$

Geometrically, this means that between a and b, there is a tangent to the graph of h that has the same slope; see Figure 2.

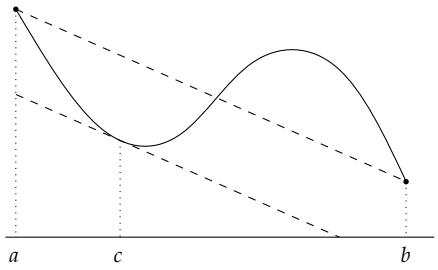


Figure 2. Illustration of the mean value theorem.

Theorem 6 (Fundamental theorem of calculus)**.** Let $a < b$ be real numbers, and let $h : \mathrm{dom}(h) \to \mathbb{R}$ be a differentiable function on an open domain $\mathrm{dom}(h) \supset [a, b]$, and such that h' is continuous on $[a, b]$. Then,

$$h(b) - h(a) = \int_a^b h'(t)dt.$$

2.2 *Convex sets*

Definition 7 (Convex set)**.** A set $\mathcal{C} \subseteq \mathbb{R}^d$ is convex if the line segment between any two points of $\mathcal{C}$ lies in $\mathcal{C}$. *I.e.*, if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$ and any λ with $0 \leq \lambda \leq 1$, we have

$$\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in \mathcal{C}.$$

Observation. Let $\mathcal{C}_i, i \in I$ be convex sets, where I is a (possibly infinite)

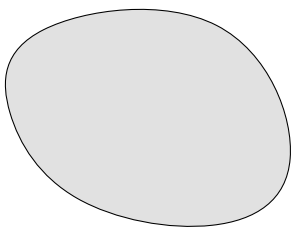
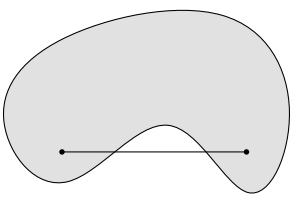


Figure 3. Example of a convex set in $\mathbb{R}^2$.



index set. Then,

$$\mathcal{C} = \bigcap_{i \in I} \mathcal{C}_i,$$

is a convex set.

2.3 *Convex functions*

Definition 8 (Convexity)**.** A function $f : \mathrm{dom}(f) \to \mathbb{R}$ is convex if (i) $\mathrm{dom}(f)$ is convex and (ii) for all $\boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}(f)$ and all $\lambda \in [0,1]$, we have

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}).$$

Geometrically, this condition means that the line segment connecting the two points $(\boldsymbol{x}, f(\boldsymbol{x}))$ and $(\boldsymbol{y}, f(\boldsymbol{y})) \in \mathbb{R}^{d+1}$ lies pointwise above the graph of f; see Figure 5.

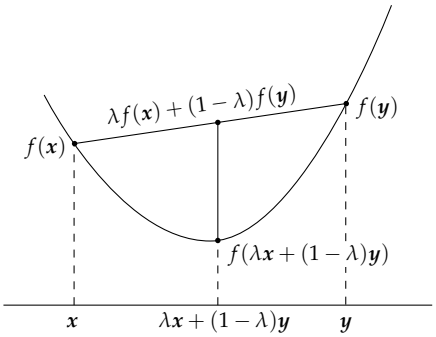


Figure 5. Illustration of the classic definition of convexity.

Definition 9 (Epigraph)**.** The epigraph of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\mathrm{epi}(f) \doteq \left\{ (\boldsymbol{x}, \alpha) \in \mathbb{R}^{d+1} \;\middle|\; \boldsymbol{x} \in \mathrm{dom}(f), \alpha \geq f(\boldsymbol{x}) \right\}.$$

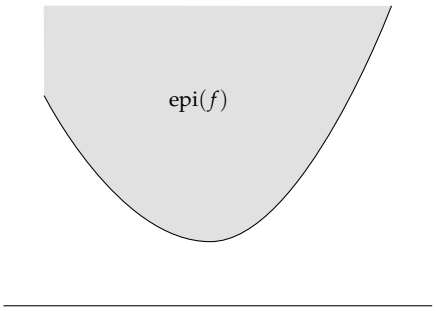


Figure 6. Epigraph of a convex function.

Observation. f is a convex function if and only if $\mathrm{epi}(f)$ is a convex set.

Lemma 10 (Jensen's inequality)**.** Let f be convex, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathrm{dom}(f)$, $\lambda_1, \ldots, \lambda_m \in \mathbb{R}_+$ such that $\sum_{i=1}^m \lambda_i = 1$, then

$$f\left(\sum_{i=1}^m \lambda_i \boldsymbol{x}_i \right) \leq \sum_{i=1}^m \lambda_i f(\boldsymbol{x}_i).$$

Proof. We will prove Jensen's inequality by induction. The base case ($m = 2$) is true by definition of convexity. Let Jensen's inequality hold for $m = k$. Consider $m = k+1$, then

$$\begin{aligned}
f\left(\sum_{i=1}^{k+1} \lambda_i \boldsymbol{x}_i \right) &= f\left(\sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i + \lambda_{k+1} \boldsymbol{x}_{k+1} \right) \\
&= f\left((1-\lambda_{k+1}) \sum_{i=1}^{k} \frac{\lambda_i}{1-\lambda_{k+1}} \boldsymbol{x}_i + \lambda_{k+1} \boldsymbol{x}_{k+1} \right) \\
&\leq (1-\lambda_{k+1}) f\left(\sum_{i=1}^{k} \frac{\lambda_i}{1-\lambda_{k+1}} \boldsymbol{x}_i \right) + \lambda_{k+1} f(\boldsymbol{x}_{k+1}) && \text{By definition of convexity.} \\
&\leq (1-\lambda_{k+1}) \sum_{i=1}^{k} \frac{\lambda_i}{1-\lambda_{k+1}} f(\boldsymbol{x}_i) + \lambda_{k+1} f(\boldsymbol{x}_{k+1}) && \text{Induction assumption, } \textstyle\sum_{i=1}^{k} \frac{\lambda_i}{1-\lambda_{k+1}} = 1. \\
&= \sum_{i=1}^{k+1} \lambda_i f(\boldsymbol{x}_i).
\end{aligned}$$

Since it holds for the base case and the induction step, Jensen's inequality must hold for all m. ∎

Lemma 11. Let f be convex and suppose that $\mathrm{dom}(f) \subseteq \mathbb{R}^d$ is open, then f is continuous.

Definition 12 (Differentiable functions)**.** Let $f : \mathrm{dom}(f) \to \mathbb{R}^m$ where $\mathrm{dom}(f) \subseteq \mathbb{R}^d$ is open. f is called differentiable at $\boldsymbol{x} \in \mathrm{dom}(f)$ if there exists an $m \times d$ matrix $\boldsymbol{A}$ and an error function $r : \mathbb{R}^d \to \mathbb{R}^m$ defined around $\mathbf{0} \in \mathbb{R}^d$ such that for all $\boldsymbol{y}$ in some neighborhood of $\boldsymbol{x}$,

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{A}(\boldsymbol{y} - \boldsymbol{x}) + r(\boldsymbol{y} - \boldsymbol{x}),$$

where $\lim_{v \to \mathbf{0}} \|r(v)\| / \|v\| = \mathbf{0}$ (error function r is sublinear around $\mathbf{0}$). $\boldsymbol{A}$ is unique and called the Jacobian matrix of f at $\boldsymbol{x}$.

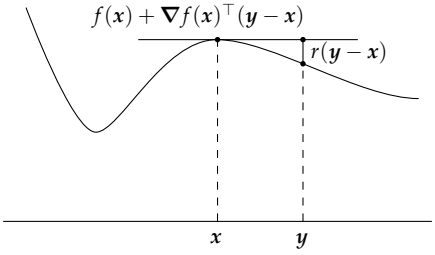


Figure 7. Graph of the affine function $f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$ is a tangent hyperplane to the graph of f at $(\boldsymbol{x}, f(\boldsymbol{x}))$.

Lemma 13 (First-order convexity)**.** Suppose that $\mathrm{dom}(f)$ is open and that f is differentiable. In particular, the gradient

$$\nabla f(\boldsymbol{x}) \doteq \left[\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial x_d} \right]$$

exists at every point $\boldsymbol{x} \in \mathrm{dom}(f)$.

Then, f is convex if and only if (i) $\mathrm{dom}(f)$ is convex and (ii)

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}(f) : f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}).$$

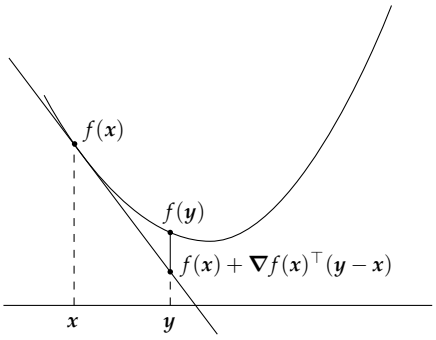


Figure 8. Illustration of the first-order characterization of convexity (Lemma 13).

Geometrically, this means that the graph is above all tangent hyperplanes; see Figure 8.

Proof. We will first prove that convexity implies that the first-order definition holds. Then, we will prove that the first-order definition implies convexity, making the definitions equivalent for differentiable functions f.

$\Rightarrow$: Suppose f is convex. Then, for all $t \in (0, 1)$,

$$\begin{aligned}
& f((1-t)\boldsymbol{x} + t\boldsymbol{y}) \leq (1-t)f(\boldsymbol{x}) + tf(\boldsymbol{y}) && \text{Definition of convexity} \\
\iff \quad & f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) \leq f(\boldsymbol{x}) + t(f(\boldsymbol{y}) - f(\boldsymbol{x})) \\
\iff \quad & f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \frac{f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - f(\boldsymbol{x})}{t} \\
& \qquad = f(\boldsymbol{x}) + \frac{\nabla f(\boldsymbol{x})^\top t(\boldsymbol{y} - \boldsymbol{x}) + r(t(\boldsymbol{y} - \boldsymbol{x}))}{t} \\
& \qquad = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \underbrace{\frac{r(t(\boldsymbol{y} - \boldsymbol{x}))}{t}}_{\to 0 \text{ for } t \to 0}.
\end{aligned}$$

$\Leftarrow$: Define $\boldsymbol{z} \doteq \lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in \mathrm{dom}(f)$ for $\lambda \in [0, 1]$ by convexity of

$\mathrm{dom}(f)$. Then, we have the following inequalities,

$$\begin{aligned} f(\boldsymbol{x}) &\geq f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{x} - \boldsymbol{z}) \\ \lambda f(\boldsymbol{x}) &\geq \lambda \Big(f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{x} - \boldsymbol{z}) \Big) \\ f(\boldsymbol{y}) &\geq f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{y} - \boldsymbol{z}) \\ (1-\lambda) f(\boldsymbol{y}) &\geq (1-\lambda) \Big(f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{y} - \boldsymbol{z}) \Big). \end{aligned}$$

This implies the following,

$$\begin{aligned} \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}) &\geq \lambda \Big(f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{x} - \boldsymbol{z}) \Big) + (1-\lambda) \Big(f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{y} - \boldsymbol{z}) \Big) \\ &= f(\boldsymbol{z}) + \lambda \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{x} - \boldsymbol{z}) + (1-\lambda) \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{y} - \boldsymbol{z}) \\ &= f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\lambda (\boldsymbol{x} - \boldsymbol{z}) + (1-\lambda)(\boldsymbol{y} - \boldsymbol{z})) \\ &= f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} - (\lambda \boldsymbol{z} + (1-\lambda)\boldsymbol{z})) \\ &= f(\boldsymbol{z}) + \boldsymbol{\nabla} f(\boldsymbol{z})^\top (\boldsymbol{z} - \boldsymbol{z}) \\ &= f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}). \end{aligned}$$

■

Lemma 14 (First-order convexity alternative)**.** Suppose that $\mathrm{dom}(f)$ is open and that f is differentiable. Then, f is convex if and only if (i) $\mathrm{dom}(f)$ is convex and (ii)

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}(f) : (\boldsymbol{\nabla} f(\boldsymbol{y}) - \boldsymbol{\nabla} f(\boldsymbol{x}))^\top (\boldsymbol{y} - \boldsymbol{x}) \geq 0.$$

Geometrically, this means that the gradient is monotonic.

Proof. We will first prove that it holds from left to right, and then from right to left.

$\Rightarrow$: If f is convex, the first-order characterization of convexity (Lemma 13) yields the following,

$$\begin{aligned} f(\boldsymbol{x}) &\geq f(\boldsymbol{y}) + \boldsymbol{\nabla} f(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) \\ f(\boldsymbol{y}) &\geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}), \end{aligned}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}(f)$. Adding up these two inequalities yields

$$\begin{aligned} & f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{y}) + \boldsymbol{\nabla} f(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) + f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) \\ \iff \quad & 0 \geq \boldsymbol{\nabla} f(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) \\ & \quad = (\boldsymbol{\nabla} f(\boldsymbol{y}) - \boldsymbol{\nabla} f(\boldsymbol{x}))^\top (\boldsymbol{x} - \boldsymbol{y}) \\ \iff \quad & 0 \leq (\boldsymbol{\nabla} f(\boldsymbol{y}) - \boldsymbol{\nabla} f(\boldsymbol{x}))^\top (\boldsymbol{y} - \boldsymbol{x}). \end{aligned}$$

$\Leftarrow$: Define $\boldsymbol{z} \doteq (1-t)\boldsymbol{x} + t\boldsymbol{y} \in \mathrm{dom}(f)$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}(f), t \in (0,1)$ by convexity of $\mathrm{dom}(f)$. Observe that $\boldsymbol{z} = \boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})$. Then, we have the

following inequality, according to the monotonicity of the gradient,

$$\begin{aligned}(\nabla f(z) - \nabla f(x))^\top (z - x) &\geq 0 \\ (\nabla f(x + t(y - x)) - \nabla f(x))^\top (x + t(y - x) - x) &\geq 0 \\ (\nabla f(x + t(y - x)) - \nabla f(x))^\top (y - x) &\geq 0. \qquad \text{Divide by } t\end{aligned}$$

Let $h(t) = f(x + t(y - x))$, then $h'(t) = \nabla f(x + t(y - x))^\top (y - x)$. Hence, we can rewrite the inequality as the following,

$$h'(t) \geq \nabla f(x)^\top (y - x).$$

By the mean value theorem, there exists $c \in (0, 1)$ such that $h'(c) = h(1) - h(0)$. *I.e.*,

$$h'(c) = f(y) - f(x).$$

Thus, we can rewrite the inequality to the following,

$$\begin{aligned}f(y) &= f(x) + h'(c) \\ &\geq f(x) + \nabla f(x)^\top (y - x),\end{aligned}$$

which is the first-order characterization of convexity (Lemma 13).

■

Lemma 15 (Second-order convexity)**.** Suppose that $\mathrm{dom}(f)$ is open and that f is twice differentiable. In particular, the Hessian

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_d \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_d^2} \end{bmatrix}$$

exists at every point $x \in \mathrm{dom}(f)$ and is symmetric.

Then, f is convex if and only if (i) $\mathrm{dom}(f)$ is convex and (ii) for all $x \in \mathrm{dom}(f)$, we have

$$\nabla^2 f(x) \succeq 0.$$

I.e., $\nabla^2 f(x)$ is positive semidefinite (M is positive semidefinite if $x^\top M x \geq 0$ for all x and $x^\top M x > 0$ for all $x \neq \mathbf{0}$).

Geometrically, this means that f has non-negative curvature everywhere. *I.e.*, the growth rate should be growing.

Observation (Operations that preserve convexity)**.** Let $f_1, \ldots, f_m$ be convex functions, $\lambda_1, \ldots, \lambda_m \in \mathbb{R}_+$, then

$$f \doteq \max_{i=1}^{m} f_i,$$

and

$$f \doteq \sum_{i=1}^{m} \lambda_i f_i.$$

are convex on $\mathrm{dom}(f) \doteq \bigcap_{i=1}^{m} \mathrm{dom}(f_i)$.

Let f be a convex function with $\text{dom}(f) \subseteq \mathbb{R}^d, g : \mathbb{R}^m \to \mathbb{R}^d$ an affine function, *i.e.* $g(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ for some $\boldsymbol{A} \in \mathbb{R}^{d\times m}, \boldsymbol{b} \in \mathbb{R}^d$. Then, the function $f \circ g$ is convex on $\text{dom}(f \circ g) \doteq \{\boldsymbol{x} \in R^m \mid g(\boldsymbol{x}) \in \text{dom}(f)\}$.

2.4 *Minimizing convex functions*

Definition 16 (Local minimum)**.** A local minimum of $f : \text{dom}(f) \to \mathbb{R}$ is a point $\boldsymbol{x}$ such that there exists $\epsilon > 0$ with

$$f(\boldsymbol{x}) \leq f(\boldsymbol{y}) \;\; \forall \boldsymbol{y} \in \text{dom}(f) \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{x}\| < \epsilon.$$

Remark. This definition of local minima means that in some small neighborhood around $\boldsymbol{x}$, $\boldsymbol{x}$ is the smallest point.

Lemma 17. Let $\boldsymbol{x}^\star$ be a local minimum of a convex function $f : \text{dom}(f) \to \mathbb{R}$. Then, $\boldsymbol{x}^\star$ is a global minimum, meaning that $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{y}) \;\; \forall \boldsymbol{y} \in \text{dom}(f)$.

Proof. Proof by contradiction. Suppose there exists $\boldsymbol{y} \in \text{dom}(f)$ such that $f(\boldsymbol{y}) < f(\boldsymbol{x}^\star)$. Define $\boldsymbol{y}' \doteq \lambda\boldsymbol{x}^\star + (1-\lambda)\boldsymbol{y}$ for $\lambda \in (0,1)$. From convexity, we get that $f(\boldsymbol{y}') < f(\boldsymbol{x}^\star)$, because $f(\boldsymbol{y}) < f(\boldsymbol{x}^\star)$. If we choose λ so close to 1 such that $\|\boldsymbol{y}' - \boldsymbol{x}^\star\| < \epsilon$ gives the inequality $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{y}')$. This yields a contradiction, thus there cannot exist a $\boldsymbol{y} \in \text{dom}(f)$ such that $f(\boldsymbol{y}) < f(\boldsymbol{x}^\star)$, meaning that $f(\boldsymbol{x}^\star) \leq f(\boldsymbol{y})$ for all $\boldsymbol{y} \in \text{dom}(f)$. ■

Lemma 18. Suppose that f is convex and differentiable over an open domain $\text{dom}(f)$. Let $\boldsymbol{x} \in \text{dom}(f)$. If $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$ (critical point), then $\boldsymbol{x}$ is a global minimum.

Proof. Suppose that $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$. According to the first-order characterization of convexity, we have

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \underbrace{\boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})}_{=0} = f(\boldsymbol{x})$$

for all $\boldsymbol{y} \in \text{dom}(f)$, so $\boldsymbol{x}$ is a global minimum. ■

Definition 19 (Strict convexity)**.** A function $f : \text{dom}(f) \to \mathbb{R}$ is strictly convex if (i) $\text{dom}(f)$ is convex and (ii) for all $\boldsymbol{x} \neq \boldsymbol{y} \in \text{dom}(f)$ and all $\lambda \in (0,1)$, we have

$$f(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) < \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}).$$

An example of a strictly convex function can be found in Figure 5, while an example of a function that is not strictly convex can be found in Figure 10.

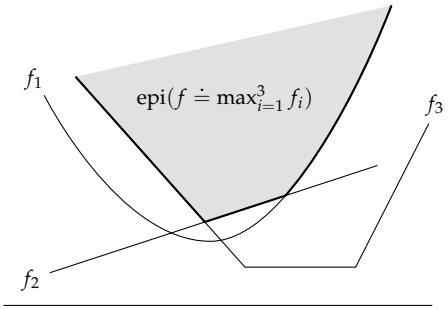


Figure 9. The maximum operator over m convex functions is a convex function. As can be seen, the epigraph of f is convex.

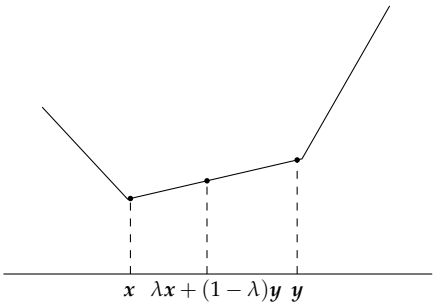


Figure 10. A non-strictly convex function.

Lemma 20. Let $f : \mathrm{dom}(f) \to \mathbb{R}$ be strictly convex. Then, f has *at most* one global minimum.

But, how do we know a minimizer exists? To be able to find this out, we must use the Weierstrass theorem.

Definition 21 (Sublevel set)**.** $f : \mathbb{R}^d \to \mathbb{R}, \alpha \in \mathbb{R}$. The set $f^{\leq \alpha} \doteq \{\boldsymbol{x} \in \mathbb{R}^d \mid f(\boldsymbol{x}) \leq \alpha\}$ is the α-sublevel set of f.

Theorem 22 (Weierstrass)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous function, and suppose there is a non-empty and bounded sublevel set $f^{\leq \alpha}$. Then, f has a global minimum.

Proof. We know that f attains a minimum over the closed and bounded set $f^{\leq \alpha}$ at some $\boldsymbol{x}^\star$, because f is continuous. This $\boldsymbol{x}^\star$ is also a global minimum as it has value $f(\boldsymbol{x}^\star) \leq \alpha$, while any $\boldsymbol{x} \notin f^{\leq \alpha}$ has value $f(\boldsymbol{x}) > \alpha \geq f(\boldsymbol{x}^\star)$. ■

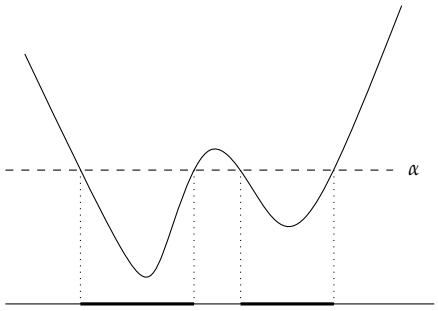


Figure 11. Illustration of a sublevel set of a non-convex function.

Since convex functions are continuous by Lemma 11, the Weierstrass theorem also applies to convex functions. Thus, to figure out whether a convex function has a minimizer, we need to find any non-empty and bounded sublevel set of this function.

2.5 *Convex programming*

In standard form, optimization problems look like the following,

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, m \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1, \ldots, p. \end{aligned}$$

The problem domain is then $\mathcal{D} = \bigcap_{i=0}^{m} \mathrm{dom}(f_i) \cap \bigcap_{j=1}^{p} \mathrm{dom}(h_j)$.

In a convex program, all f_i are convex functions, and all h_j are affine functions with domain $\mathbb{R}^d$.

Definition 23 (Lagrange dual function)**.** Given an optimization problem in standard form, its Lagrangian is the function $L : \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ given by

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) + \sum_{j=1}^{p} \nu_j h_j(\boldsymbol{x}).$$

The λ_i, ν_j are called Lagrange multipliers. The Lagrange dual function is the function $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R} \cup \{-\infty\}$ defined by

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\boldsymbol{x} \in \mathcal{D}} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}).$$

Only the $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ pairs with $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) > -\infty$ are interesting.

Lemma 24 (Weak Lagrange duality)**.** Let $\boldsymbol{x}$ be a feasible solution ($f_i(\boldsymbol{x}) \leq 0$ for $i = 1, \dots m$ and $h_j(\boldsymbol{x}) = 0$ for $j = 1, \dots, p$). Let g be the Lagrange dual function and $\boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\nu} \in \mathbb{R}^p$ such that $\boldsymbol{\lambda} \geq \mathbf{0}$. Then

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq f_0(\boldsymbol{x}).$$

Proof.

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\boldsymbol{x}) + \underbrace{\sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x})}_{\leq 0} + \underbrace{\sum_{j=1}^{p} \nu_j h_j(\boldsymbol{x})}_{=0} \leq f_0(\boldsymbol{x}).$$

■

However, we want to know what the best lower bound is that we can get in this way. For this, we must choose $\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\nu}$ such that $g(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is maximized. This can be phrased as another optimization problem, called the Lagrange dual,

$$\begin{aligned} \text{maximize} \quad & g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned}$$

Observation. The Lagrange dual is a convex program, even if the primal is not.

Theorem 25. Suppose that a convex program has a feasible solution $\tilde{\boldsymbol{x}}$ that in addition satisfies $f_i(\tilde{\boldsymbol{x}}) < 0, i = 1, \dots, m$ (Slater point). Then, the infimum value of the primal equals the supremum value of its Lagrange dual. Moreover, if the value is finite, it is attained by a feasible solution of the dual,

$$\inf f_0(\boldsymbol{x}) = \max g(\boldsymbol{\lambda}, \boldsymbol{\nu}).$$

A case of particular interest is that strong duality holds and the joint value is attained in both the primal and dual problem.[1] If all f_i and h_j are differentiable, then the Karush-Kuhn-Tucker (KKT) conditions provide necessary and, under convexity, also sufficient conditions for this case to occur.

[1] This would mean that $\min f_0(\boldsymbol{x}) = \max g(\boldsymbol{\lambda}, \boldsymbol{\nu})$.

Definition 26 (Zero duality gap)**.** Let $\tilde{\boldsymbol{x}}$ be feasible for the primal and $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ feasible for the Lagrange dual. The primal and dual solutions $\tilde{\boldsymbol{x}}$ and $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ are said to have zero duality gap if $f_0(\tilde{\boldsymbol{x}}) = g(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$.

If the primal and dual have zero duality gap, then $\min f_0(\boldsymbol{x}) = \max g(\boldsymbol{\lambda}, \boldsymbol{\nu})$.

The consequence of zero duality gap is the *master equation*,

$$\begin{aligned} f_0(\tilde{\boldsymbol{x}}) &= g(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}}) \\ &= \inf_{\boldsymbol{x} \in \mathcal{D}} \left(f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \tilde{\lambda}_i f_i(\boldsymbol{x}) + \sum_{j=1}^{p} \tilde{\nu}_j h_j(\boldsymbol{x}) \right) \\ &\leq f_0(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m} \underbrace{\tilde{\lambda}_i f_i(\tilde{\boldsymbol{x}})}_{\leq 0} + \sum_{j=1}^{p} \underbrace{\tilde{\nu}_j h_j(\tilde{\boldsymbol{x}})}_{=0} \\ &\leq f_0(\tilde{\boldsymbol{x}}), \end{aligned}$$

which means that all inequalities turn into equalities. Thus,

$$\tilde{\lambda}_i f_i(\tilde{\boldsymbol{x}}) = 0, \quad i = 1, \ldots, m,$$

which is called complementary slackness, because if $\tilde{\lambda}_i \neq 0$, then $f_i(\tilde{\boldsymbol{x}}) = 0$, and vice versa. Furthermore, if all f_i and h_j are differentiable, then

$$\boldsymbol{\nabla} f_0(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m} \tilde{\lambda}_i \boldsymbol{\nabla} f_i(\tilde{\boldsymbol{x}}) + \sum_{j=1}^{p} \tilde{\nu}_j \boldsymbol{\nabla} h_j(\tilde{\boldsymbol{x}}) = \mathbf{0},$$

which is called the vanishing Lagrangian gradient condition.

In summary, we get the following result.

Theorem 27 (Karush-Kuhn-Tucker necessary conditions)**.** Let $\tilde{\boldsymbol{x}}$ and $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\nu}})$ be feasible solutions of the primal optimization problem and its Lagrange dual, respectively, with zero duality gap. If all f_i and h_j are differentiable, then

$$\tilde{\lambda}_i f_i(\tilde{\boldsymbol{x}}) = 0, \quad i = 1, \ldots, m \qquad \text{Complementary slackness}$$

$$\boldsymbol{\nabla} f_0(\tilde{\boldsymbol{x}}) + \sum_{i=1}^{m} \tilde{\lambda}_i \boldsymbol{\nabla} f_i(\tilde{\boldsymbol{x}}) + \sum_{j=1}^{p} \tilde{\nu}_j \boldsymbol{\nabla} h_j(\tilde{\boldsymbol{x}}) = \mathbf{0}. \qquad \text{Vanishing Lagrangian gradient}$$

If we have a convex program, then the KKT conditions are sufficient for zero duality gap. The motivation behind this is that they may be easier to solve for than solving the primal optimization problem. However, we cannot always count on the KTT conditions to be solvable, because they are only guaranteed if there are primal and dual solutions of zero duality gap. But, if the primal has a Slater point[2], then the KTT conditions are equivalent to the existence of a primal minimizer.

[2] A point $\tilde{\boldsymbol{x}}$ such that $f_i(\tilde{\boldsymbol{x}}) < 0$ for all $i = 1, \ldots, m$.

3 *Gradient descent*

Gradient descent is an optimization algorithm that aims to find the global minimizer. We assume that $f : \mathbb{R}^d \to \mathbb{R}$ is convex, differentiable, and has a global minimizer $x^\star$. Then, the goal of gradient descent is, for $\epsilon > 0$, to find a $x \in \mathbb{R}^d$ such that

$$f(x) - f(x^\star) \leq \epsilon.$$

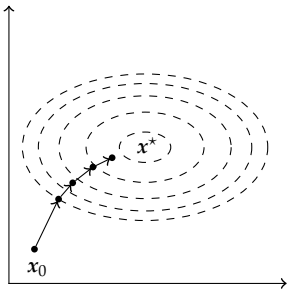


Figure 12. Gradient descent updates.

It works by first choosing a $x_0 \in \mathbb{R}^d$ and then iteratively updating this value by

$$x_{t+1} = x_t - \gamma \nabla f(x_t),$$

for timesteps $t = 0, 1, \ldots$ and stepsize $\gamma > 0$.

Take a small step into the direction of the negative gradient to move toward the minimum.

3.1 *Vanilla analysis*

We want to be able to bound $f(x_t) - f(x^\star)$, which tells us how far off we are from the minimum at timestep t. For this, we can use the first-order characterization of convexity,

$$f(x^\star) \geq f(x_t) + \nabla f(x_t)^\top (x^\star - x_t) \iff f(x_t) - f(x^\star) \leq \nabla f(x_t)^\top (x_t - x^\star).$$

Using the gradient descent update rule, we get

$$\nabla f(x_t) = \frac{x_t - x_{t+1}}{\gamma},$$

which gives the following bound

$$\begin{aligned}
f(x_t) - f(x^\star) &\leq \nabla f(x_t)^\top (x_t - x^\star) \\
&= \frac{1}{\gamma}(x_t - x_{t+1})^\top (x_t - x^\star) \\
&= \frac{1}{2\gamma}\Big(\|x_t - x_{t+1}\|^2 + \|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2\Big) && 2v^\top w = \|v\|^2 + \|w\|^2 - \|v - w\|^2 \text{ (cosine theorem).} \\
&= \frac{\gamma}{2}\|\nabla f(x_t)\|^2 + \frac{1}{2\gamma}\Big(\|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2\Big). && x_t - x_{t+1} = \gamma \nabla f(x_t).
\end{aligned}$$

Summing this up over the first T iterations, we get an upper bound on the summed error,

$$\begin{aligned}
\sum_{t=0}^{T-1} (f(x_t) - f(x^\star)) &\leq \sum_{t=0}^{T-1} \nabla f(x_t)^\top (x_t - x^\star) \\
&= \frac{\gamma}{2} \sum_{t=0}^{T-1} \|\nabla f(x_t)\|^2 + \frac{1}{2\gamma}\Big(\|x_0 - x^\star\|^2 - \|x_T - x^\star\|^2\Big) && \text{Telescoping sum.} \\
&\leq \frac{\gamma}{2} \sum_{t=0}^{T-1} \|\nabla f(x_t)\|^2 + \frac{1}{2\gamma}\|x_0 - x^\star\|^2. && \|x_T - x^\star\|^2 \geq 0.
\end{aligned}$$

We can also use this to get a bound on the average error,

$$\frac{1}{T}\sum_{t=0}^{T-1} (f(x_t) - f(x^\star)) \leq \frac{1}{T}\left(\frac{\gamma}{2} \sum_{t=0}^{T-1} \|\nabla f(x_t)\|^2 + \frac{1}{2\gamma}\|x_0 - x^\star\|^2\right).$$

3.2 *Lipschitz convex functions*

Theorem 28. Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\boldsymbol{x}^\star$. Furthermore, suppose that $\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\| \leq R$ and $\|\nabla f(\boldsymbol{x})\| \leq B$ for all $\boldsymbol{x}$. Choosing the stepsize,

$$\gamma \doteq \frac{R}{B\sqrt{T}},$$

gradient descent yields

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

A function $f : \mathbb{R}^m \to \mathbb{R}^n$ is Lipschitz continuous if there exists an M, such that $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}$:

$$\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \leq M\|\boldsymbol{x} - \boldsymbol{y}\|.$$

This holds if and only if the gradient is bounded.

Proof. We can plug our bounds into the vanilla analysis,

$$\begin{aligned}\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) &\leq \frac{\gamma}{2}\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{x}_t)\|^2 + \frac{1}{2\gamma}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 \\ &\leq \frac{\gamma}{2}B^2T + \frac{1}{2\gamma}R^2 \\ &\doteq q(\gamma).\end{aligned}$$

Now, we want to choose γ such that the bound $q(\gamma)$ is minimized. We can compute the minimum by computing the derivative,

$$q'(\gamma) = \frac{1}{2}B^2T - \frac{R^2}{2\gamma^2},$$

and solving for $q'(\gamma) = 0$, which yields

$$\gamma = \frac{R}{B\sqrt{T}}.$$

Then, we can compute the bound by

$$q\left(\frac{R}{B\sqrt{T}}\right) = RB\sqrt{T}.$$

Dividing by T yields the result. ∎

We want to find out how many iterations we would need to ensure that the average error is bounded by ϵ. We can use Theorem 28 to compute a lower bound on the number of iterations,

$$\frac{RB}{\sqrt{T}} \leq \epsilon \implies T \geq \frac{R^2B^2}{\epsilon^2}.$$

So, the amount of iterations until convergence is of the order $\mathcal{O}(1/\epsilon^2)$. This means that we need at most $10000 \cdot R^2B^2$ iterations to achieve an error of 0.01.

3.3 *Smooth functions*

Definition 29 (Smoothness)**.** Let $f : \mathrm{dom}(f) \to \mathbb{R}$ be differentiable, $\mathcal{X} \subseteq \mathrm{dom}(f)$ convex, $L \in \mathbb{R}_+$. f is smooth (with parameter L) over $\mathcal{X}$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$:

$$\underbrace{f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})}_{\text{first-order convexity}} + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

Remark. This definition does not require convexity.

Geometrically, this definition of smoothness means that the graph f is below a not too steep tangent paraboloid at $(\boldsymbol{x}, f(\boldsymbol{x}))$; see Figure 13. Furthermore, we can easily check if f is smooth with parameter L if the following function is convex,

$$g(\boldsymbol{x}) \doteq \frac{L}{2}\boldsymbol{x}^\top \boldsymbol{x} - f(\boldsymbol{x}),$$

over $\mathrm{dom}(g) \doteq \mathrm{dom}(f)$.

Observation (Operations that preserve smoothness)**.** Let $f_1, \ldots, f_m$ be smooth with parameters $L_1, \ldots, L_m$ and let $\lambda_1, \ldots, \lambda_m \in \mathbb{R}_+$. Then, the function $f \doteq \sum_{i=1}^m \lambda_i f_i$ is smooth with parameter $\sum_{i=1}^m \lambda_i L_i$.

Let f be smooth with parameter L and let $g(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$. Then, the function $f \circ g$ is smooth with parameter $L\|\boldsymbol{A}\|^2$, where $\|\boldsymbol{A}\|$ is the spectral norm of $\boldsymbol{A}$.

Lemma 30. Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable, then f is smooth with parameter L if and only if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$:

$$\|\boldsymbol{\nabla} f(\boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|.$$

I.e., $\boldsymbol{\nabla} f$ is Lipschitz continuous.

Lemma 31 (Sufficient decrease)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and smooth with parameter L. With stepsize

$$\gamma \doteq \frac{1}{L},$$

gradient descent satisfies

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{1}{2L}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2.$$

Sufficient decrease implies that, every step, we get closer to the minimum.

"Not too curved."

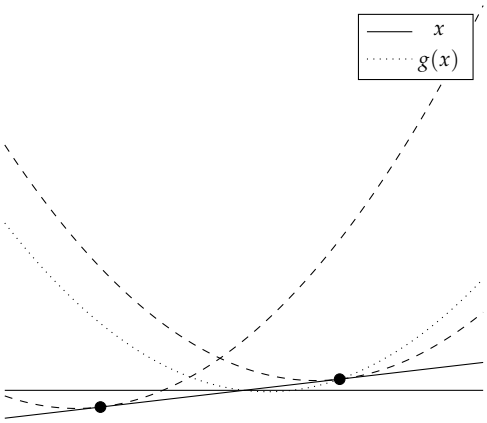


Figure 13. Plot of $f(x) = x$ with the tangent paraboloids at $x = -8, 5$, showing smoothness with parameter $L = 1$. Furthermore, it shows that $g(x) = \frac{1}{2}x^2 - f(x)$ is convex.

Proof.

$$
\begin{aligned}
f(\boldsymbol{x}_{t+1}) &\leq f(\boldsymbol{x}_t) + \boldsymbol{\nabla} f(\boldsymbol{x}_t)^\top (\boldsymbol{x}_{t+1} - \boldsymbol{x}_t) + \frac{L}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 && \text{Smoothness.} \\
&= f(\boldsymbol{x}_t) + \boldsymbol{\nabla} f(\boldsymbol{x}_t)^\top \left(-\frac{\boldsymbol{\nabla} f(\boldsymbol{x}_t)}{L}\right) + \frac{L}{2}\left\|\frac{\boldsymbol{\nabla} f(\boldsymbol{x}_t)}{L}\right\|^2 && \text{GD: } \boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \tfrac{1}{L}\boldsymbol{\nabla} f(\boldsymbol{x}_t). \\
&= f(\boldsymbol{x}_t) - \frac{1}{L}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 + \frac{1}{2L}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 \\
&= f(\boldsymbol{x}_t) - \frac{1}{2L}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2.
\end{aligned}
$$

■

Theorem 32. Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\boldsymbol{x}^\star$. Furthermore, suppose that f is smooth with parameter L. Choosing stepsize

$$\gamma \doteq \frac{1}{L},$$

gradient descent yields

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \leq \frac{L}{2T}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2, \quad T > 0.$$

Proof. Due to sufficient decrease, we can bound the sum of squared gradients (which will be useful when bounding the vanilla analysis),

$$
\begin{aligned}
\frac{1}{2L}\sum_{t=0}^{T-1}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 &\leq \sum_{t=0}^{T-1} f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) && \text{SD: } f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \tfrac{1}{2L}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2. \\
&= f(\boldsymbol{x}_0) - f(\boldsymbol{x}_T). && \text{Telescoping sum.}
\end{aligned}
$$

Using the vanilla analysis, we can derive a bound on the average error,

$$
\begin{aligned}
\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) &\leq \frac{1}{2L}\sum_{t=0}^{T-1}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 + \frac{L}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 && \gamma \doteq \tfrac{1}{L}. \\
&\leq f(\boldsymbol{x}_0) - f(\boldsymbol{x}_T) + \frac{L}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 \\
\sum_{t=1}^{T}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) &\leq \frac{L}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 \\
\frac{1}{T}\sum_{t=1}^{T}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) &\leq \frac{L}{2T}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2.
\end{aligned}
$$

From sufficient decrease, we know that the last iterate must be the best, thus

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \leq \frac{1}{T}\sum_{t=1}^{T}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) \leq \frac{L}{2T}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2.$$

■

Using this result, we can compute a bound on T to achieve an error smaller than ϵ,

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \leq \frac{L}{2T}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 \leq \frac{R^2 L}{2T} \leq \epsilon.$$

The error then becomes,

$$T \geq \frac{R^2 L}{2\epsilon},$$

which is on the order of $\mathcal{O}(1/\epsilon)$. This means that we need at most $50 \cdot R^2 L$ iterations for an error of 0.01 (as opposed to $10000 \cdot R^2 B^2$ in the Lipschitz case).

3.4 *Smooth and strongly convex functions*

Definition 33 (Strong convexity)**.** Let $f : \mathrm{dom}(f) \to \mathbb{R}$ be a convex and differentiable function, $\mathcal{X} \in \mathrm{dom}(f)$ convex and $\mu \in \mathbb{R}_+, \mu > 0$. f is called *strongly convex* with parameter μ over $\mathcal{X}$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

Geometrically, this means that, for any $\boldsymbol{x}$, the graph of f is above a not too flat tangent paraboloid at $(\boldsymbol{x}, f(\boldsymbol{x}))$; see Figure 14. Furthermore, we can easily check if f is strongly convex if and only if the following function is convex,

$$g(\boldsymbol{x}) \doteq f(\boldsymbol{x}) - \frac{\mu}{2}\boldsymbol{x}^\top \boldsymbol{x}.$$

By assuming that a function f is smooth and strongly convex, we can use a stronger lower bound to derive a bound on the error from the vanilla analysis.

"Not too flat."

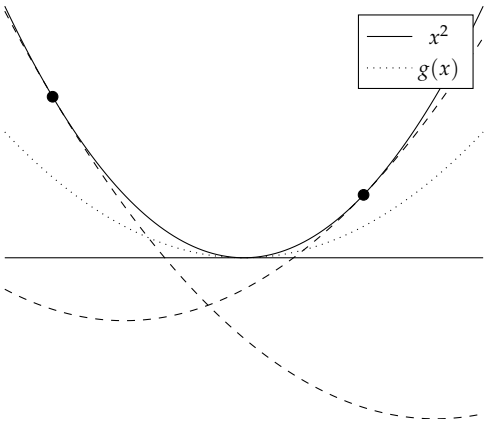


Figure 14. Plot of $f(x) = x^2$ with the tangent paraboloids at $x = -6, 4$, showing strong convexity with parameter $\mu = 1$. Furthermore, it shows $g(x) = f(x) - \frac{1}{2}x^2$, which is convex.

Theorem 34. Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\boldsymbol{x}^\star$. Furthermore, suppose that f is smooth with parameter L and strongly convex with parameter $\mu > 0$. Choosing $\gamma \doteq 1/L$, gradient descent with arbitrary $\boldsymbol{x}_0$ satisfies the following two properties,

1. Squared distances to $\boldsymbol{x}^\star$ are geometrically decreasing,

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \leq \left(1 - \frac{\mu}{L}\right)\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2;$$

2. The absolute error after T iterations is exponentially small in T,

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \leq \frac{L}{2}\left(1 - \frac{\mu}{L}\right)^T \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2.$$

Proof of 1. Using the vanilla analysis, we know the following,

$$\begin{aligned} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) &\leq \boldsymbol{\nabla} f(\boldsymbol{x}_t)^\top (\boldsymbol{x}_t - \boldsymbol{x}^\star) \\ &= \frac{\gamma}{2} \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 + \frac{1}{2\gamma} \Big(\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \Big). \end{aligned}$$

But, using strong convexity, we have a stronger lower bound on the left hand side,

$$\begin{aligned} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) &\leq \boldsymbol{\nabla} f(\boldsymbol{x}_t)^\top (\boldsymbol{x}_t - \boldsymbol{x}^\star) - \frac{\mu}{2} \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 && \text{Strong convexity.} \\ &= \frac{\gamma}{2} \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 + \frac{1}{2\gamma} \Big(\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \Big) \\ &\quad - \frac{\mu}{2} \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2. \end{aligned}$$

This can be rewritten to the following bound,

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \leq \underbrace{2\gamma(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}_t)) + \gamma^2 \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2}_{\text{“noise“}} + (1 - \mu\gamma) \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2.$$

Now we need to show that the noise is non-positive,

$$\begin{aligned} 2\gamma(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}_t)) + \gamma^2 \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 &= \frac{2}{L}(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}_t)) + \frac{1}{L^2} \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 && \gamma \doteq \tfrac{1}{L}. \\ &\leq \frac{2}{L}(f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t)) + \frac{1}{L^2} \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 && \text{Sufficient decrease.} \\ &\leq -\frac{1}{L^2} \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 + \frac{1}{L^2} \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 && \text{SD: } f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \tfrac{1}{2L} \|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2. \\ &= 0. \end{aligned}$$

Hence, the noise is non-positive, and we get the following,

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \leq \Big(1 - \frac{\mu}{L}\Big) \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2.$$

■

Proof of 2. From (1), we know the following,

$$\|\boldsymbol{x}_T - \boldsymbol{x}^\star\|^2 \leq \Big(1 - \frac{\mu}{L}\Big)^T \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2.$$

Using smoothness, we can derive a bound on the final iterate error,

$$\begin{aligned} f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) &\leq \boldsymbol{\nabla} f(\boldsymbol{x}^\star)^\top (\boldsymbol{x}_T - \boldsymbol{x}^\star) + \frac{L}{2} \|\boldsymbol{x}_T - \boldsymbol{x}^\star\|^2 \\ &= \frac{L}{2} \|\boldsymbol{x}_T - \boldsymbol{x}^\star\|^2 && \boldsymbol{\nabla} f(\boldsymbol{x}^\star) = 0, \text{ because it is a stationary point.} \\ &\leq \frac{L}{2} \Big(1 - \frac{\mu}{L}\Big)^T \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2. && \text{Using (1).} \end{aligned}$$

■

From there, we can derive a lower bound on the number of iterations T to get an error of at most ϵ,

$$T \geq \frac{L}{\mu} \log\Big(\frac{R^2 L}{2\epsilon}\Big),$$

This means that we need $\frac{L}{\mu}\log(50 \cdot R^2 L)$ iterations for an error of at most 0.01, as opposed to $50 \cdot R^2 L$ in the smooth case. This bound only depends linearly on $\frac{\mu}{L}$, which might be very high.

4 *Projected gradient descent*

In constrained optimization, we want to minimize $f(\boldsymbol{x})$, subject to $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$. We will assume that $\mathcal{X} \subsetneq \mathbb{R}^d$ is a closed convex set. The idea of projected gradient descent is to project onto $\mathcal{X}$ after every step,

$$\Pi_{\mathcal{X}}(\boldsymbol{y}) \doteq \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x}-\boldsymbol{y}\|^2.$$

The algorithm thus alternates between the following two steps,

$$\begin{aligned}\boldsymbol{y}_{t+1} &\doteq \boldsymbol{x}_t - \gamma\nabla f(\boldsymbol{x}_t)\\ \boldsymbol{x}_{t+1} &\doteq \Pi_{\mathcal{X}}(\boldsymbol{y}_{t+1}).\end{aligned}$$

Observation. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be closed and convex, $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathbb{R}^d$, then

1. $(\boldsymbol{x} - \Pi_{\mathcal{X}}(\boldsymbol{y}))^\top(\boldsymbol{y} - \Pi_{\mathcal{X}}(\boldsymbol{y})) \leq 0$;
2. $\|\boldsymbol{x} - \Pi_{\mathcal{X}}(\boldsymbol{y})\|^2 + \|\boldsymbol{y} - \Pi_{\mathcal{X}}(\boldsymbol{y})\|^2 \leq \|\boldsymbol{x}-\boldsymbol{y}\|^2$.

The two properties equivalently say that the vectors $\boldsymbol{y} - \Pi_{\mathcal{X}}(\boldsymbol{y})$ and $\boldsymbol{y} - \Pi_{\mathcal{X}}(\boldsymbol{y})$ form an obtuse angle; see Figure 15.

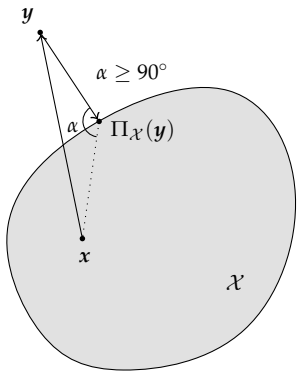


Figure 15. Proof by picture of the properties of the projection step made in projected gradient descent.

4.1 *Lipschitz convex functions*

Since we minimize f over a closed and bounded convex set $\mathcal{X}$, we get the existence of a minimizer and bound R for free. Furthermore, if we assume that f is continuously differentiable, we also have a bound B for the gradient norms over $\mathcal{X}$. This makes the following theorem a much more useful result than in the unconstrained case, since we need to make less assumptions about f.

Theorem 35. Let $f : \operatorname{dom}(f) \to \mathbb{R}$ be convex and differentiable, $\mathcal{X} \subseteq \operatorname{dom}(f)$ closed and convex, $\boldsymbol{x}^\star$ a minimizer of f over $\mathcal{X}$. Furthermore, suppose that $\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\| \leq R$ and $\|\nabla f(\boldsymbol{x})\| \leq B$ for all $\boldsymbol{x} \in \mathcal{X}$. Choosing the constant stepsize,

$$\gamma \doteq \frac{R}{B\sqrt{T}},$$

with projected gradient descent yields the following bound,

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

Proof. We only need to show that the vanilla analysis still holds for the projected $\boldsymbol{x}_{t+1}$. We know that for the non-projected iterate, we get the following equality from the vanilla analysis,

$$\nabla f(\boldsymbol{x}_t)^\top(\boldsymbol{x}_t - \boldsymbol{x}^\star) = \frac{1}{2\gamma}\Big(\gamma^2\|\nabla f(\boldsymbol{x}_t)\|^2 + \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{y}_{t+1} - \boldsymbol{x}^\star\|^2\Big).$$

By the second property of projected gradient descent, we know $\|x_{t+1} - x^\star\|^2 \leq \|y_{t+1} - x^\star\|^2$, hence we get

$$\nabla f(x_t)^\top (x_t - x^\star) \leq \frac{1}{2\gamma}\Big(\gamma^2 \|\nabla f(x_t)\|^2 + \|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2\Big).$$

Then, we can prove the rest of the vanilla analysis as for unbounded gradient descent. ■

This is the same bound as in the unconstrained case, thus the number of necessary iteration is of the order $\mathcal{O}(1/\epsilon^2)$.

4.2 *Smooth functions*

Lemma 36 (Sufficient decrease of projected gradient descent)**.** Let $f : \mathrm{dom}(f) \to \mathbb{R}$ be differentiable and smooth with parameter L over a closed and convex set $\mathcal{X} \subseteq \mathrm{dom}(f)$. Choosing stepsize

$$\gamma \doteq \frac{1}{L},$$

projected gradient descent satisfies

$$f(x_{t+1}) \leq f(x_t) - \frac{1}{2L}\|\nabla f(x_t)\|^2 + \underbrace{\frac{L}{2}\|y_{t+1} - x_{t+1}\|^2}_{\text{additional term}}.$$

Proof.

$$\begin{aligned}
f(x_{t+1}) &\leq f(x_t) + \nabla f(x_t)^\top (x_{t+1} - x_t) + \frac{L}{2}\|x_t - x_{t+1}\|^2 && \text{Smoothness.}\\
&= f(x_t) - L(y_{t+1} - x_t)^\top (x_{t+1} - x_t) + \frac{L}{2}\|x_t - x_{t+1}\|^2 && y_{t+1} = x_t - \tfrac{1}{L}\nabla f(x_t) \Rightarrow \nabla f(x_t) = -L(y_{t+1} - x_t).\\
&= f(x_t) - \frac{L}{2}\Big(\|y_{t+1} - x_t\|^2 + \|x_{t+1} - x_t\|^2 - \|y_{t+1} - x_{t+1}\|\Big) && 2v^\top w = \|v\|^2 + \|w\|^2 - \|v - w\|^2.\\
&\quad + \frac{L}{2}\|x_t - x_{t+1}\|^2 \\
&= f(x_t) - \frac{L}{2}\|y_{t+1} - x_t\|^2 + \frac{L}{2}\|y_{t+1} - x_{t+1}\|^2 \\
&= f(x_t) - \frac{1}{2L}\|\nabla f(x_t)\|^2 + \frac{L}{2}\|y_{t+1} - x_{t+1}\|^2. && y_{t+1} = x_t - \tfrac{1}{L}\nabla f(x_t) \Rightarrow y_{t+1} - x_t = -\tfrac{1}{L}\nabla f(x_t).
\end{aligned}$$

■

Thus, we also have a sufficient decrease lemma for this version of gradient descent, which has an additional term in its bound. However, as we will see, this does not matter, because we can compensate for it in the vanilla analysis.

Theorem 37. Let $f : \mathrm{dom}(f) \to \mathbb{R}$ be convex and differentiable. Let $\mathcal{X} \subseteq \mathrm{dom}(f)$ be a closed convex set, and $x^\star$ the minimizer of f over $\mathcal{X}$. Furthermore, suppose that f is smooth over $\mathcal{X}$ with parameter L. Choosing stepsize

$$\gamma \doteq \frac{1}{L},$$

projected gradient descent yields the following bound,

$$f(x_T) - f(x^\star) \leq \frac{L}{2T}\|x_0 - x^\star\|^2.$$

Proof. From property 2 of gradient descent, we get the following inequality,

$$\|x_{t+1} - x^\star\|^2 + \|y_{t+1} - x_{t+1}\|^2 \leq \|y_{t+1} - x^\star\|^2.$$

Using this inequality, we get the following upper bound,

$$\begin{aligned} f(x_t) - f(x^\star) &\leq \nabla f(x_t)^\top (x_t - x^\star) \\ &= \frac{1}{2\gamma}\Big(\gamma^2\|\nabla f(x_t)\|^2 + \|x_t - x^\star\|^2 - \|y_{t+1} - x^\star\|^2\Big) && \text{See vanilla analysis, where } x_{t+1} \text{ is substituted by } y_{t+1}\text{, since that is the next unconstrained iterate.} \\ &\leq \frac{1}{2\gamma}\Big(\gamma^2\|\nabla f(x_t)\|^2 + \|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2 \\ &\qquad - \|y_{t+1} - x_{t+1}\|^2\Big). \end{aligned}$$

We use sufficient decrease to bound the sum of gradients,

$$\begin{aligned} \frac{1}{2L}\sum_{t=0}^{T-1}\|\nabla f(x_t)\|^2 &\leq \sum_{t=0}^{T-1} f(x_t) - f(x_{t+1}) + \frac{L}{2}\|y_{t+1} - x_{t+1}\|^2 && \text{Sufficient decrease.} \\ &= f(x_0) - f(x_T) + \frac{L}{2}\sum_{t=0}^{T-1}\|y_{t+1} - x_{t+1}\|^2. && \text{Telescoping sum.} \end{aligned}$$

Now, we can bound the summed error and we will see that the additional terms cancel,

$$\begin{aligned} \sum_{t=0}^{T-1}(f(x_t) - f(x^\star)) &\leq \frac{1}{2L}\sum_{t=0}^{T-1}\|\nabla f(x_t)\|^2 + \frac{L}{2}\|x_0 - x^\star\|^2 \\ &\qquad - \frac{L}{2}\sum_{t=0}^{T-1}\|y_{t+1} - x_{t+1}\|^2 \\ &\leq f(x_0) - f(x_T) + \frac{L}{2}\sum_{t=0}^{T-1}\|y_{t+1} - x_{t+1}\|^2 \\ &\qquad + \frac{L}{2}\|x_0 - x^\star\|^2 - \frac{L}{2}\sum_{t=0}^{T-1}\|y_{t+1} - x_{t+1}\|^2 \\ &= f(x_0) - f(x_T) + \frac{L}{2}\|x_0 - x^\star\|^2, \end{aligned}$$

which results in

$$\sum_{t=1}^{T}(f(x_t) - f(x^\star)) \leq \frac{L}{2}\|x_0 - x^\star\|^2.$$

Due to sufficient decrease, we get

$$f(x_T) - f(x^\star) \leq \frac{L}{2T}\|x_0 - x^\star\|^2.$$

■

This is the same bound as in the unconstrained case, thus the number of necessary iterations is of the order $\mathcal{O}(1/\epsilon)$.

4.3 *Smooth and strongly convex functions*

Theorem 38. Let $f : \mathrm{dom}(f) \to \mathbb{R}$ be convex and differentiable. Let $\mathcal{X} \subseteq \mathrm{dom}(f)$ be a nonempty closed and convex set and suppose that f is smooth over $\mathcal{X}$ with parameter L and strongly convex over $\mathcal{X}$ with parameter $\mu > 0$. Choosing stepsize

$$\gamma \doteq \frac{1}{L},$$

projected gradient descent satisfies the following two properties,

1. Squared distance to $x^\star$ are geometrically decreasing,

$$\|x_{t+1} - x^\star\|^2 \leq \left(1 - \frac{\mu}{L}\right)\|x_t - x^\star\|^2;$$

2. The absolute error after T iterations is exponentially small in T,

$$f(x_T) - f(x^\star) \leq \frac{L}{2}\left(1 - \frac{\mu}{L}\right)^T \|x_0 - x^\star\|^2 + \underbrace{\|\nabla f(x^\star)\|\left(1 - \frac{\mu}{L}\right)^{T/2}\|x_0 - x^\star\|}_{\text{additional term}}.$$

Proof. These proofs are similar to the normal gradient descent case, starting from the constrained vanilla bound,

$$\frac{1}{2\gamma}\left(\gamma^2\|\nabla f(x_t)\|^2 + \|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2 - \|y_{t+1} - x_{t+1}\|^2\right),$$

which can be strengthened to

$$\frac{1}{2\gamma}\left(\gamma^2\|\nabla f(x_t)\|^2 + \|x_t - x^\star\|^2 - \|x_{t+1} - x^\star\|^2 - \|y_{t+1} - x_{t+1}\|^2\right) - \frac{\mu}{2}\|x_t - x^\star\|^2,$$

on $f(x_t) - f(x^\star)$. ■

5 *Coordinate descent*

A problem with gradient descent in large-scale learning is that we need to compute the full gradient $\nabla f(\boldsymbol{x}_t)$ in every iteration. The idea of *coordinate descent* is to update only one coordinate of the iterate at a time. To do this, we only need to compute one coordinate of $\nabla f(\boldsymbol{x}_t)$ at a time, which we assume to be a factor of d faster to compute.

However, we also expect to pay a price for this in terms of the number of iterations required. It turns out that, in the worst case, the number of iterations will increase by a factor of d, so we only stand to gain. Under additional assumptions about f, coordinate descent can lead to provable speedups.

Definition 39 (Polyak-Łojasiewicz (PL) inequality)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function with a global minimum $\boldsymbol{x}^\star$. We say that f satisfies the PL inequality if the following holds for some $\mu > 0$,

$$\frac{1}{2}\|\nabla f(\boldsymbol{x})\|^2 \geq \mu(f(\boldsymbol{x}) - f(\boldsymbol{x}^\star)), \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

Observation. The PL inequality directly implies that every critical point is a minimizer of f.

There are non-convex functions that satisfy the PL inequality.

Lemma 40. Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and strongly convex with parameter $\mu > 0$. Then f satisfies the PL inequality for the same μ.

To analyze coordinate descent, we need the notion of *coordinate-wise smoothness*.

Definition 41 (Coordinate-wise smoothness)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable, and $\mathcal{L} = [L_1, \ldots, L_d] \in \mathbb{R}^d_+$. Function f is called coordinate-wise smooth (with parameter $\mathcal{L}$) if for every coordinate $i = 1, \ldots, d$, the following holds,

$$f(\boldsymbol{x} + \lambda \boldsymbol{e}_i) \leq f(\boldsymbol{x}) + \lambda \nabla_i f(\boldsymbol{x}) + \frac{L_i}{2}\lambda^2, \quad \forall \boldsymbol{x} \in \mathbb{R}^d, \lambda \in \mathbb{R}.$$

Compare this to the definition of smoothness; Definition 29. In our new coordinate-wise definition, we define $\boldsymbol{y}$ as $\boldsymbol{x} + \lambda \boldsymbol{e}_i$, since we only want to change coordinate i. Hence, $\boldsymbol{y} - \boldsymbol{x}$ becomes $\lambda \boldsymbol{e}_i$. From there, it is easy to that $\nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$ becomes $\lambda \nabla_i f(\boldsymbol{x})$, and $\|\boldsymbol{x} - \boldsymbol{y}\|$ becomes λ. Thus, smoothness with parameter L implies coordinate-wise smoothness with parameter $\mathcal{L} = [L, \ldots, L]$.

Example 42. $f(x_1, x_2) = x_1^2 + 10x_2^2$ is smooth with parameter $L = 20$, but f is coordinate-wise smooth with parameter $\mathcal{L} = [2, 20]$. Such differences will become important later when showing faster convergence of coordinate descent.

In general, coordinate (gradient) descent algorithms perform the following actions,

$$\begin{aligned} &\text{choose an active coordinate } i \in [d] \\ &\boldsymbol{x}_{t+1} \doteq \boldsymbol{x}_t - \gamma_i \boldsymbol{\nabla}_i f(\boldsymbol{x}_t) \boldsymbol{e}_i. \end{aligned}$$

We call the coordinate that is currently being update *active*.

Lemma 43 (Coordinate-wise sufficient decrease)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and coordinate-wise smooth with parameter $\mathcal{L} = [L_1, \ldots, L_2]$. With active coordinate i in iteration t and stepsize

$$\gamma_i \doteq \frac{1}{L_i},$$

coordinate descent satisfies

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{1}{2L_i} |\boldsymbol{\nabla}_i f(\boldsymbol{x}_t)|^2.$$

Proof. Let $\lambda = -\boldsymbol{\nabla}_i f(\boldsymbol{x}_t)/L_i$, then $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \lambda \boldsymbol{e}_i$. Then, we can apply coordinate-wise smoothness,

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) &\leq f(\boldsymbol{x}_t) + \lambda \boldsymbol{\nabla}_i f(\boldsymbol{x}_t) + \frac{L_i}{2} \lambda^2 \\ &= f(\boldsymbol{x}_t) - \frac{1}{L_i} |\boldsymbol{\nabla}_i f(\boldsymbol{x}_t)|^2 + \frac{1}{2L_i} |\boldsymbol{\nabla}_i f(\boldsymbol{x}_t)|^2 \\ &= f(\boldsymbol{x}_t) - \frac{1}{2L_i} |\boldsymbol{\nabla}_i f(\boldsymbol{x}_t)|^2. \end{aligned}$$

Coordinate-wise smoothness.

■

5.1 Randomized coordinate descent

In *randomized gradient descent*, the active coordinate is chosen uniformly at random,

$$\begin{aligned} &\text{sample } i \in [d] \text{ uniformly at random} \\ &\boldsymbol{x}_{t+1} \doteq \boldsymbol{x}_t - \gamma_i \boldsymbol{\nabla}_i f(\boldsymbol{x}_t) \boldsymbol{e}_i. \end{aligned}$$

Randomized coordinate descent is at least as fast as gradient descent on smooth functions if we assume that it is d times cheaper to update one coordinate than the full iterate [Nesterov, 2012]. If we additionally assume the PL inequality, we can obtain faster convergence.

Theorem 44. Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with global minimum $x^\star$. Suppose that f is coordinate-wise smooth with parameter L and satisfies the PL inequality with parameter $\mu > 0$. Choosing stepsize

$$\gamma_i \doteq \frac{1}{L},$$

randomized gradient descent with arbitrary x_0 yields

$$\mathbb{E}[f(x_T) - f(x^\star)] \leq \left(1 - \frac{\mu}{dL}\right)^T (f(x_0) - f(x^\star)).$$

Proof. Sufficient decrease yields

$$f(x_{t+1}) \leq f(x_t) - \frac{1}{2L} |\nabla_i f(x_t)|^2.$$

By taking the expectation of both sides with respect to i, we have

$$\begin{aligned}
\mathbb{E}_i[f(x_{t+1}) \mid x_t] &\leq \mathbb{E}_i\left[f(x_t) - \frac{1}{2L}|\nabla_i f(x_t)|^2 \;\middle|\; x_t\right] \\
&= f(x_t) - \frac{1}{2L}\sum_{i=1}^{d} \frac{1}{d} |\nabla_i f(x_t)|^2 \\
&= f(x_t) - \frac{1}{2dL}\|\nabla f(x_t)\|^2 \\
&\leq f(x_t) - \frac{\mu}{dL}(f(x_t) - f(x^\star)). && \text{PL inequality.}
\end{aligned}$$

Subtracting $f(x^\star)$ from both sides and taking the expectation over x_t, we obtain

$$\mathbb{E}[f(x_{t+1}) - f(x^\star)] \leq \left(1 - \frac{\mu}{dL}\right)(f(x_t) - f(x^\star)).$$

$\mathbb{E}_i[f(x_{t+1}) \mid x_t]$ is a random variable whose expectation is $\mathbb{E}_{x_t}[f(x_{t+1})]$.

The statement follows. ∎

5.2 *Importance sampling*

As seen, uniformly random selection of the active coordinate does not yield a better bound than gradient descent. However, we have not made use of the fact that the coordinate-wise smoothness parameters can differ. Intuitively, we would want to sample coordinates with high smoothness more frequently than coordinates with low smoothness, since they change more rapidly, leading to faster convergence to the optimum. This leads us to *importance sampling* [Nesterov, 2012],

$$\begin{aligned}
&\text{sample } i \in [d] \text{ with probability } \frac{L_i}{\sum_{j=1}^d L_j} \\
&x_{t+1} \doteq x_t - \gamma_i \nabla_i f(x_t) e_i.
\end{aligned}$$

Theorem 45. Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $x^\star$. Suppose that f is coordinate-wise smooth with parameter $\mathcal{L} = [L_1, \dots, L_d]$ and satisfies the PL inequality with parameter $\mu > 0$. Let

$$\overline{L} = \frac{1}{d} \sum_{i=1}^{d} L_i$$

be the average of coordinate-wise smoothness constants. Then, coordinate descent with importance sampling and arbitrary x_0 yields

$$\mathbb{E}[f(x_T - f(x^\star)] \leq \left(1 - \frac{\mu}{d\overline{L}}\right)^T (f(x_0) - f(x^\star)).$$

Proof. The proof is nearly identical to the proof of randomized coordinate descent. The difference lies in the expectation over i. Importance sampling yields

$$\begin{aligned}
\mathbb{E}_i\left[f(x_t) - \frac{1}{2L_i}|\nabla_i f(x_t)|^2 \,\middle|\, x_t\right] &= f(x_t) - \sum_{i=1}^{d} \frac{L_i}{\sum_{j=1}^{d} L_j} \frac{1}{2L_i} |\nabla_i f(x_t)|^2 \\
&= f(x_t) - \frac{1}{2d\overline{L}} \sum_{i=1}^{d} |\nabla_i f(x_t)|^2. \qquad d\overline{L} = \textstyle\sum_{i=1}^{d} L_i.
\end{aligned}$$

The rest follows identically. ∎

Note that $\overline{L}$ can be much smaller than $L = \max_{i=1}^{d} L_i$, so coordinate descent with important sampling is potentially faster than randomized gradient descent. In the worst-case, both algorithms are the same.

5.3 *Steepest coordinate descent*

In contrast to random coordinate descent, *steepest coordinate descent* chooses the active coordinate according to the coordinate with the largest gradient (*Gauss-Southwell* rule),

$$\begin{aligned}
&\text{choose } i = \operatorname*{argmax}_{i \in [d]} |\nabla_i f(x_t)| \\
&x_{t+1} \doteq x_t - \gamma_i \nabla_i f(x_t) e_i.
\end{aligned}$$

The main difference from the previous algorithms is that this algorithm is deterministic, thus we do not need to take the expectation.

Corollary. Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $x^\star$. Suppose that f is coordinate-wise smooth with parameter L and satisfies the PL inequality with parameter $\mu > 0$. Choosing stepsize

$$\gamma_i \doteq \frac{1}{L},$$

steepest coordinate descent with arbitrary x_0 yields

$$f(x_T) - f(x^\star) \leq \left(1 - \frac{\mu}{dL}\right)^T (f(x_0) - f(x^\star)).$$

This is not good. It needs the same amount of iterations as randomized coordinate descent, but each iteration takes as long as in gradient descent.[3] However, this algorithm allows for a speedup in certain cases. Furthermore, it may be possible to efficiently maintain the maximum absolute gradient value throughout the iterations, so that the full evaluation of the gradient can be avoided.

[3] Note that a function may be coordinate-wise smooth with an L for all coordinates that is smaller than smoothness, so it is not completely fair to compare this to gradient descent.

Nutini et al. [2015] showed that a better convergence result can be obtained for strongly convex functions, when strong convexity is measured w.r.t. the ℓ_1 norm instead of the standard Euclidean norm, *i.e.*,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu_1}{2}\|y - x\|_1^2.$$

Due to $\|y - x\|_1 \geq \|y - x\|$, f is then also strongly convex with $\mu = \mu_1$. On the other hand, if f is μ-strongly convex w.r.t. the ℓ_2 norm, then f is μ/d-strongly convex w.r.t. the ℓ_1 norm, due to $\|y - x\| \geq \|y-x\|_1/\sqrt{d}$.

Lemma 46. Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and strongly convex with parameter $\mu_1 > 0$ w.r.t. the ℓ_1 norm. Then, f is μ_1-strongly convex w.r.t. the Euclidean norm, so a global minimum $x^\star$ exists. Then, f satisfies the PL inequality w.r.t. the ℓ_∞ norm with the same μ_1,

$$\frac{1}{2}\|\nabla f(x)\|_\infty^2 \geq \mu_1 (f(x) - f(x^\star)).$$

Theorem 47. Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $x^\star$. Suppose that f is coordinate-wise smooth with parameter L and satisfies the PL inequality with parameter $\mu_1 > 0$. Choosing stepsize

$$\gamma_i \doteq \frac{1}{L},$$

steepest coordinate descent with arbitrary x_0 yields

$$f(x_T) - f(x^\star) \leq \left(1 - \frac{\mu_1}{L}\right)^T (f(x_0) - f(x^\star)).$$

Proof.

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) &\leq f(\boldsymbol{x}_t) - \frac{1}{2L}|\boldsymbol{\nabla}_i f(\boldsymbol{x}_t)|^2 && \text{Sufficient decrease.} \\ &= f(\boldsymbol{x}_t) - \frac{1}{2L}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|_\infty^2 && \text{Active coordinate of steepest coordinate descent is the maximum gradient.} \\ &\leq f(\boldsymbol{x}_t) - \frac{\mu_1}{L}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)). && \text{PL inequality.} \end{aligned}$$

Subtracting $f(\boldsymbol{x}^\star)$ from both sides yields

$$f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}^\star) \leq \left(1 - \frac{\mu_1}{L}\right)(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)).$$

The statement follows. ■

We see that if $\mu_1 = \mu/d$, we do not gain anything. However, this is not the case in general. If, for the worst case $\boldsymbol{x}, \boldsymbol{y}$, which satisfy $\|\boldsymbol{y} - \boldsymbol{x}\| = \|\boldsymbol{y}-\boldsymbol{x}\|_1/\sqrt{d}$, strong convexity holds with $\mu' > \mu$, then we can achieve $\mu_1 = \mu'/d > \mu/d$, resulting in better convergence.

5.4 *Greedy coordinate descent*

Greedy coordinate descent is a variant that does not even require f to be differentiable. In each iteration, we make the step that maximizes the progress in the chosen coordinate. This requires performing a *line search* by solving a one-dimensional optimization problem,

$$\begin{aligned} &\text{choose } i \in [d] \\ &\boldsymbol{x}_{t+1} \doteq \operatorname*{argmin}_{\lambda \in \mathbb{R}} f(\boldsymbol{x}_t + \lambda \boldsymbol{e}_i). \end{aligned}$$

However, greedy coordinate descent can get stuck in non-optimal points; see Figure 16. Thus, we need some additional conditions to make sure this does not occur.

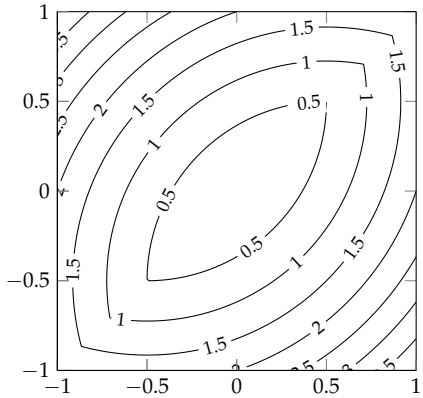


Figure 16. Level set plot of $f(\boldsymbol{x}) = \|\boldsymbol{x}\|^2 + |x_1 - x_2|$. The global minimum is $[0, 0]$, but greedy coordinate descent cannot escape any point $[x, x]$, s.t. $|x| \leq 1/2$.

Theorem 48. Let $f : \mathbb{R}^d \to \mathbb{R}$ be of the form

$$f(\boldsymbol{x}) \doteq g(\boldsymbol{x}) + h(\boldsymbol{x}), \quad h(\boldsymbol{x}) = \sum_{i=1}^{d} h_i(x_i), \quad \boldsymbol{x} \in \mathbb{R}^d,$$

with g convex and differentiable, and the h_i convex.

Let $\boldsymbol{x}$ be a point such that greedy coordinate descent cannot make progress in any coordinate. Then $\boldsymbol{x}$ is a global minimum of f.

In the context of machine learning, an important class of functions that satisfies the conditions of Theorem 48 is the following form,

$$f(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_1,$$

where $\lambda\|\boldsymbol{x}\|_1$ is a separable ℓ_1-regularization term used in for example LASSO [Tibshirani, 1996].

6 Nonconvex functions

So far, all convergence results that we have proved have been for variants of gradient descent on convex functions. The reason for this is that, in general, we cannot expect gradient descent to come close to the global minimum $x^\star$ of nonconvex functions. Figures 17 to 19 show what can go wrong in nonconvex functions, under the assumption that we have set the γ such that we do not overshoot. These figures show points that gradient descent cannot escape; local minima and saddle points. Furthermore, it might even be that gradient descent never converges to a critical point; see Figure 19.

In practice, gradient descent works well on the nonconvex functions that we care about. But, theoretical explanations for this are mostly missing. Despite this, we will show that under favorable conditions, we can still say something useful about the behavior of gradient descent on nonconvex functions.

We can easily make an analysis of gradient descent on smooth functions. A useful property that we will use is that functions with bounded Hessians are smooth, as shown in the following lemma.

Lemma 49. Let $f : \mathrm{dom}(f) \to \mathbb{R}$ be twice differentiable with $\mathcal{X} \in \mathrm{dom}(f)$ a convex set and $\|\nabla^2 f(x)\| \leq L$ for all $x \in \mathcal{X}$. Then f is smooth with parameter L over $\mathcal{X}$.

Proof. Bounded Hessians imply Lipschitz continuity of the gradient,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{X}.$$

We will use the fundamental theorem of calculus with

$$h(t) \doteq f(x + t(y - x)), \quad t \in [0, 1].$$

The derivative can be calculated by chain rule,

$$h'(t) = \nabla f(x + t(y - x))^\top (y - x).$$

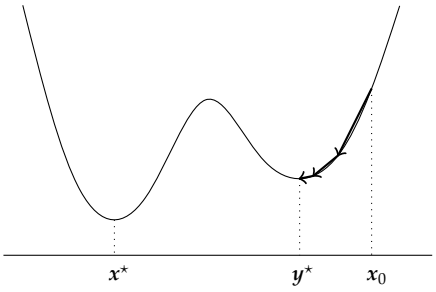


Figure 17. Gradient descent may get stuck in a local minimum $y^\star \neq x^\star$, in nonconvex functions.

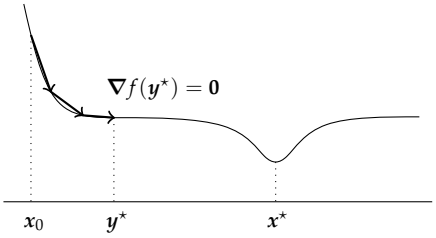


Figure 18. Gradient may get stuck in a flat region (saddle point) in nonconvex functions.

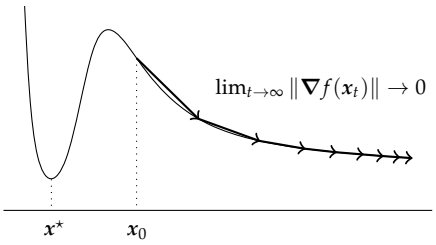


Figure 19. Gradient descent may never even reach a critical point in nonconvex functions.

Now, we can show smoothness,

$$
\begin{aligned}
& f(\boldsymbol{y}) - f(\boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) \\
&= h(1) - h(0) - \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) && \text{Definition of } h. \\
&= \int_0^1 h'(t) \mathrm{d}t - \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) && \text{Fundamental theorem of calculus.} \\
&= \int_0^1 \boldsymbol{\nabla} f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))^\top (\boldsymbol{y} - \boldsymbol{x}) \mathrm{d}t - \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) && \text{Fill in } h'(t). \\
&= \int_0^1 \boldsymbol{\nabla} f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x}))^\top (\boldsymbol{y} - \boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) \mathrm{d}t && \text{Integral from 0 to 1.} \\
&= \int_0^1 (\boldsymbol{\nabla} f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \boldsymbol{\nabla} f(\boldsymbol{x}))^\top (\boldsymbol{y} - \boldsymbol{x}) \mathrm{d}t \\
&\leq \int_0^1 \left| (\boldsymbol{\nabla} f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \boldsymbol{\nabla} f(\boldsymbol{x}))^\top (\boldsymbol{y} - \boldsymbol{x}) \right| \mathrm{d}t \\
&\leq \int_0^1 \| \boldsymbol{\nabla} f(\boldsymbol{x} + t(\boldsymbol{y} - \boldsymbol{x})) - \boldsymbol{\nabla} f(\boldsymbol{x}) \| \| \boldsymbol{y} - \boldsymbol{x} \| \mathrm{d}t && \text{Cauchy-Schwarz inequality.} \\
&\leq \int_0^1 L \| t(\boldsymbol{y} - \boldsymbol{x}) \| \| \boldsymbol{y} - \boldsymbol{x} \| \mathrm{d}t && \text{Lipschitz continuous gradient.} \\
&= \int_0^1 L t \| \boldsymbol{x} - \boldsymbol{y} \|^2 \mathrm{d}t \\
&= \frac{L}{2} \| \boldsymbol{x} - \boldsymbol{y} \|^2.
\end{aligned}
$$

Thus, we have smoothness,

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

■

Now, we can use this fact and sufficient decrease[4] to prove that the gradients of smooth functions are bounded and approach o, as we increase the number of iterations.

[4] Recall that sufficient decrease did not require convexity.

Theorem 50. Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $\boldsymbol{x}^\star$. Furthermore, suppose that f is smooth with parameter L. Choosing stepsize

$$\gamma \doteq \frac{1}{L},$$

gradient descent yields

$$\frac{1}{T} \sum_{t=0}^{T-1} \| \boldsymbol{\nabla} f(\boldsymbol{x}_t) \|^2 \leq \frac{2L}{T} (f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star)).$$

Remark. Note that concave functions are not a counter example to this theorem, despite their gradients growing, because they have no global minimum $\boldsymbol{x}^\star$.

Proof. Recall that sufficient decrease does not require convexity,

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{1}{2L} \| \boldsymbol{\nabla} f(\boldsymbol{x}_t) \|^2.$$

Rewriting this, we get

$$\|\nabla f(\boldsymbol{x}_t)\|^2 \leq 2L(f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1})).$$

Then, by telescoping sum, we get

$$\sum_{t=0}^{T-1} \|\nabla f(\boldsymbol{x}_t)\|^2 \leq 2L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}_T)) \leq 2L(f(\boldsymbol{x}_0) - f(\boldsymbol{x}^\star)).$$

The statement follows by dividing both sides by T. ■

This has the result that

$$\lim_{t\to\infty} \|\nabla f(\boldsymbol{x}_t)\|^2 = 0.$$

It might seem that convergence of the gradients to 0 is the same as convergence to a critical point. But, this interpretation does not hold in general; see Figure 19. In this case, the gradient converges to 0, but the iterates only move further away from the critical point. So, this is not a very strong result.

6.1 *Trajectory analysis*

Despite the fact that a nonconvex function may contain local minima, saddle points, and flat parts, gradient descent may avoid them and still converge to a global minimum. For this, you need a good starting point and do a trajectory analysis. As an example, we will do a trajectory analysis for a simplified deep linear neural network. It turns out that this function is smooth along the trajectories that we analysis, and this is the most important ingredient of the analysis.

Note that stacking linear layers has no benefit, since any stacking of linear layers can be represented by a single linear layer. However, the reason for doing this is that it gives us a simple playground in which we can try to understanding why training deep neural networks with gradient descent works, despite the fact that the objective function is nonconvex.

Let

$$\boldsymbol{\theta} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_\ell\}$$

be the weights of the deep linear network. And, in general, we want to approximate a matrix $\boldsymbol{Y}$, given input matrix $\boldsymbol{X}$. Thus, we want to minimize

$$\|\boldsymbol{W}_\ell \boldsymbol{W}_{\ell-1} \cdots \boldsymbol{W}_1 \boldsymbol{X} - \boldsymbol{Y}\|_F^2.$$

Arora et al. [2018] consider this general framework, but we will only consider the case where all matrices are 1×1, *i.e.*, scalars. Assume we have training input $x = 1$ and output $y = 1$, then we have the following function to optimize,

$$f(\boldsymbol{x}) \doteq \frac{1}{2}\left(\prod_{k=1}^{d} x_k - 1\right)^2.$$

We rewrite $\boldsymbol{w}$ as $\boldsymbol{x}$ and ℓ as d to be more in line with the notation used here.

We can immediately see that setting $x_k = 1$ for all k minimizes the function at 0. However, we want to know whether gradient descent will also be able to find this set of weights.

The gradient of this function is computed by

$$\nabla_i f(\boldsymbol{x}) = \left(\prod_{k=1}^{d} x_k - 1\right) \prod_{k \neq i}^{d} x_k$$

Whenever at least two dimensions are zero, the gradient vanishes. Thus, any $\boldsymbol{x}$ with two zero entries are critical points, despite not being global minima, since then the product of all entries must be 1, which is not possible if at least two are zero.[5] We know that the value of all such saddle points is $1/2$.

[5] This shows that f is nonconvex, since local minima are global minima in convex functions.

We now want to show that for any number of layers (*i.e.* dimensionality of $\boldsymbol{x}$), anywhere in $\mathcal{X} = \left\{\boldsymbol{x} \;\middle|\; \boldsymbol{x} > \mathbf{0}, \prod_{k=1}^{d} x_k \leq 1\right\}$, despite that f is not smooth over $\mathcal{X}$. However, we only need to show that f is smooth along the trajectory of gradient descent for suitable L, so that we get sufficient decrease. We will now show this by showing that the Hessians over the trajectory are bounded. The Hessian is given by

$$\nabla^2_{ij} f(\boldsymbol{x}) = \begin{cases} \left(\prod_{k \neq i}^{d} x_k\right)^2, & j = i \\ 2 \prod_{k \neq i}^{d} x_k \prod_{k \neq j}^{d} x_k - \prod_{k \neq i,j}^{d} x_k, & j \neq i. \end{cases}$$

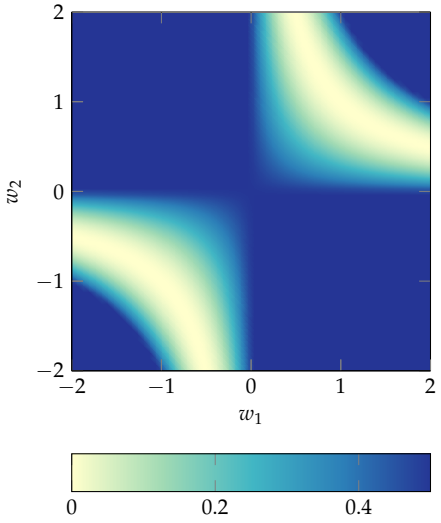


Figure 20. $f(\boldsymbol{x}) = \frac{1}{2}(\prod_k x_k - 1)^2$ for $d = 2$, where the loss is clipped to be at most $1/2$.

Definition 51 (c-balanced.)**.** Let $\boldsymbol{x} > \mathbf{0}$ and $c \geq 1$. $\boldsymbol{x}$ is called c-balanced if $x_i \leq c x_j$ for all $1 \leq i, j \leq d$.

Lemma 52. Let $\boldsymbol{x} > \mathbf{0}$ be c-balanced with $\prod_k x_k \leq 1$, then for any stepsize $\gamma > 0$, $\boldsymbol{x}' \doteq \boldsymbol{x} - \gamma \nabla f(\boldsymbol{x})$ satisfies $\boldsymbol{x}' \geq \boldsymbol{x}$ componentwise, and is also c-balanced.

Proof. Let

$$\Delta \doteq -\gamma \left(\prod_{k=1}^{d} x_k - 1\right) \left(\prod_{k=1}^{d} x_k\right) \geq 0.$$

Then,

$$-\gamma \nabla_k f(\boldsymbol{x}) = \frac{\Delta}{x_k}.$$

Thus, the gradient descent update has the following form,

$$x'_k = x_k + \frac{\Delta}{x_k} \geq x_k, \quad k \in [d].$$

For all i, j, we thus get

$$x'_i = x_i + \frac{\Delta}{x_i} \leq c x_j + \frac{c\Delta}{x_j} = c x'_j.$$

$x_j \leq c x_i \iff \frac{1}{x_i} \leq \frac{c}{x_j}$.

■

So, we know now that all iterates are c-balanced if $\boldsymbol{x}_0$ is balanced with c. We can use this to compute a bound on the Hessian by bounding the products.

Lemma 53. Suppose $\boldsymbol{x} > \mathbf{0}$ is c-balanced. Then, for any $I \subseteq [d]$, we have

$$\prod_{k \notin I} x_k \leq c^{|I|} \left(\prod_{k=1}^{d} x_k \right)^{1-|I|/d} \leq c^{|I|}.$$

Proof. For any $i \in [d]$, we have $c^d \cdot x_i^d \geq \prod_k x_k$ by c-balancedness, hence $x_i \geq \frac{1}{c} (\prod_k x_k)^{1/d}$. Thus,

$$\prod_{k \notin I} x_k = \frac{\prod_k x_k}{\prod_{i \in I} x_i} \leq \frac{\prod_k x_k}{(1/c)^{|I|} (\prod_k x_k)^{|I|/d}} = c^{|I|} \left(\prod_{k=1}^{d} x_k \right)^{1-|I|/d}.$$

Since $\prod_k x_k \leq 1$, we can bound the above by $c^{|I|}$. ■

Lemma 54. Let $\boldsymbol{x} > \mathbf{0}$ be c-balanced with $\prod_k x_k \leq 1$, then

$$\left\| \boldsymbol{\nabla}^2 f(\boldsymbol{x}) \right\| \leq \left\| \boldsymbol{\nabla}^2 f(\boldsymbol{x}) \right\|_F \leq 3dc^2.$$

Proof. The fact that $\|\boldsymbol{A}\| \leq \|\boldsymbol{A}\|_F$ is well known. To bound the Frobenius norm, we use the previous lemma to compute

$$\left| \boldsymbol{\nabla}^2_{ii} f(\boldsymbol{x}) \right| = \left| \left(\prod_{k \neq i} x_i \right)^2 \right| \leq c^2,$$

and for $i \neq j$, we get

$$\left| \boldsymbol{\nabla}^2_{ij} f(\boldsymbol{x}) \right| \leq \left| 2 \prod_{k \neq i} x_k \prod_{k \neq j} x_k \right| + \left| \prod_{k \neq i,j} x_k \right| \leq 3c^2.$$

Thus,

$$\|\boldsymbol{\nabla}^2 f(\boldsymbol{x})\|_F^2 \leq 9d^2c^4.$$

Taking the square root, the statement follows. ■

This lemma implies smoothness of f with parameter $L = 3dc^2$ along the whole trajectory of gradient descent, under the "smooth stepsize" $\gamma \doteq 1/L = 1/3dc^2$. Now, we can use this to prove convergence.

Theorem 55. Let $c \geq 1$ and $\delta < 0$ such that $\boldsymbol{x}_0 > \mathbf{0}$ is c-balanced with $\delta \leq \prod_k (\boldsymbol{x}_0)_k < 1$. Choosing stepsize

$$\gamma \doteq \frac{1}{3dc^2},$$

gradient descent satisfies

$$f(\boldsymbol{x}_T) \leq \left(1 - \frac{\delta^2}{3c^4} \right)^T f(\boldsymbol{x}_0).$$

Proof. For each $t \geq 0$, f is smooth over $\mathrm{conv}(\{\boldsymbol{x}_t, \boldsymbol{x}_{t+1}\})$ with parameter $L = 3dc^2$, hence we have sufficient decrease,

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{1}{6dc^2}\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2.$$

For every c-balanced $\boldsymbol{x}$ with $\delta \leq \prod_k x_k \leq 1$, we have

$$\begin{aligned}
\|\boldsymbol{\nabla} f(\boldsymbol{x}_t)\|^2 &= 2f(\boldsymbol{x}) \sum_{i=1}^{d} \left(\prod_{k \neq i} x_k \right)^2 \\
&\geq 2f(\boldsymbol{x}) \frac{d}{c^2} \left(\prod_{k=1}^{d} x_k \right)^{2-2/d} \\
&\geq 2f(\boldsymbol{x}) \frac{d}{c^2} \left(\prod_{k=1}^{d} x_k \right)^{2} \\
&\geq 2f(\boldsymbol{x}) \frac{d}{c^2} \delta^2.
\end{aligned}$$

Hence,

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{1}{6dc^2} 2f(\boldsymbol{x}_t) \frac{d}{c^2} \delta^2 = f(\boldsymbol{x}_t) \left(1 - \frac{\delta^2}{3c^4} \right).$$

■

Thus, we seem to have fast convergence, since the function value goes down by a constant factor in each step. However, there is a catch. Consider the $\boldsymbol{x}_0 = [1/2, \ldots, 1/2]$, which is c-balanced with $c = 1$, and $\delta = 1/2^d$. Hence, the constant factor is

$$1 - \frac{1}{3 \cdot 4^d}.$$

This means that we would need $T \approx 4^d$ iterations to reduce the initial error by a constant factor not depending on d. Hence, for this starting value, the gradient is exponentially small. In order to get polynomial convergence, we need to start with a δ that decays at most polynomially with d. For large d, this has the consequence that we must start very close to optimality. In particular, we need to start at a distance $\mathcal{O}(1/\sqrt{d})$ from the optimal solution $[1, \ldots, 1]$.

7 The Frank-Wolfe algorithm

Projected gradient descent is the only algorithm we have seen that dealt with constrained optimization problems. However, that algorithm came with the clear disadvantage that projections can be very expensive, even when $\mathcal{X}$ is convex. The Frank-Wolfe algorithm solves constrained optimization problems without projection steps. Instead, it makes use of a *linear minimization oracle* (LMO). For the feasible region $\mathcal{X} \subseteq \mathcal{R}^d$ and an arbitrary vector $g \in R^d$,[6]

$$\mathrm{LMO}_{\mathcal{X}}(g) \doteq \operatorname*{argmin}_{z \in \mathcal{X}} g^\top z.$$

Notice that this is the minimization of a linear function.

The Frank-Wolfe algorithm iteratively updates by calling the oracle in the direction of the gradient,

$$\begin{aligned} s &\doteq \mathrm{LMO}_{\mathcal{X}}(\nabla f(x_t)) \\ x_{t+1} &\doteq (1-\gamma_t)x_t + \gamma_t s. \end{aligned}$$

The algorithm reduces non-linear constrained optimization to linear optimization over the same set $\mathcal{X}$; it is able to solve general non-linear constrained optimization problems by only solving a simpler linear constrained optimization over the same set $\mathcal{X}$ in each iteration, by calling the oracle. We solve this linear optimization problem in the direction of the gradient, since that is the best linear approximation of f at x_t.

A nice property of the oracle is that if $\mathcal{X} = \mathrm{conv}(\mathcal{A})$, then $\mathrm{LMO}_{\mathcal{X}}(x) \in \mathcal{A}$. So, if we have a set $\mathcal{X}$ that is the convex hull of a small number of points, such as the ℓ_1-ball, we have an easy optimization problem with runtime $\mathcal{O}(|\mathcal{A}|)$.

The advantages of this method are

- Iterates are always feasible if $\mathcal{X}$ is convex;
- No projections, which are often harder to compute than linear optimization problems;
- Iterates always have a simple sparse representations: x_t is a convex combination of x_0 and all s used so far.

7.1 Linear minimization oracles

LASSO. The LASSO problem in its standard form is given by

$$\begin{aligned} &\text{minimize} && \|Ax - b\|^2 \\ &\text{subject to} && \|x\|_1 \leq 1. \end{aligned}$$

The constraint set $\mathcal{X} = \{x \in \mathbb{R}^d \mid \|x\|_1 \leq 1\}$ is the unit ℓ_1-ball. This is the convex hull of the unit basis vectors; $\mathcal{X} = \mathrm{conv}(\{\pm e_1, \ldots, \pm e_d\})$. The

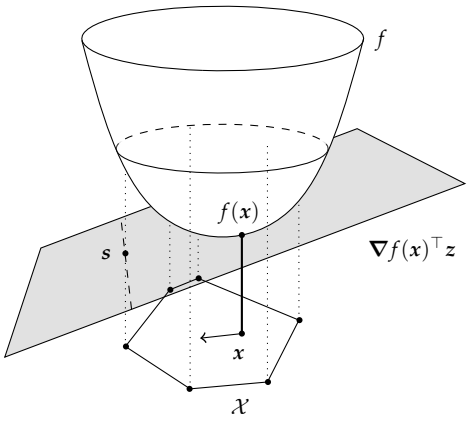


Figure 21. Illustration of a Frank-Wolfe step.

[6] g can be thought of as an "optimization direction".

Move toward minimizer.

LMO for this set is easy to compute,

$$\begin{aligned}\mathrm{LMO}_{\mathcal{X}}(\boldsymbol{g}) &= \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{X}} \boldsymbol{z}^\top \boldsymbol{g} \\ &= \operatorname*{argmin}_{\boldsymbol{z}\in\{\pm\boldsymbol{e}_1,\dots,\pm\boldsymbol{e}_d\}} \boldsymbol{z}^\top \boldsymbol{g} \\ &= -\mathrm{sign}(g_i)\boldsymbol{e}_i, \quad i \doteq \operatorname*{argmax}_{i\in[d]} |g_i|.\end{aligned}$$

So, we only have to identify $\boldsymbol{g}$'s largest coordinate, which is much more efficient than projection onto an ℓ_1-ball.

7.2 *Duality gap*

We define the duality gap of $\boldsymbol{x} \in \mathcal{X}$ as

$$g(\boldsymbol{x}) \doteq \nabla f(\boldsymbol{x})^\top (\boldsymbol{x} - \boldsymbol{s}), \quad \boldsymbol{s} \doteq \mathrm{LMO}_{\mathcal{X}}(\nabla f(\boldsymbol{x})).$$

This can be interpreted as the optimality gap $\nabla f(\boldsymbol{x})^\top \boldsymbol{x} - \nabla f(\boldsymbol{x})^\top \boldsymbol{s}$ of the linear subproblem.

Lemma 56. Suppose that the constrained minimization problem has a minimizer $\boldsymbol{x}^\star$. Let $\boldsymbol{x} \in \mathcal{X}$, then

$$g(\boldsymbol{x}) \geq f(\boldsymbol{x}) - f(\boldsymbol{x}^\star),$$

meaning that the duality gap is an upper bound for the optimality gap.

Proof.

$$\begin{aligned}g(\boldsymbol{x}) &= \nabla f(\boldsymbol{x})^\top (\boldsymbol{x} - \boldsymbol{s}) \\ &\geq \nabla f(\boldsymbol{x})^\top (\boldsymbol{x} - \boldsymbol{x}^\star) \\ &\geq f(\boldsymbol{x}) - f(\boldsymbol{x}^\star).\end{aligned}$$

$\boldsymbol{s}$ is the minimizer of $f(\boldsymbol{z}) \doteq \nabla f(\boldsymbol{x})^\top \boldsymbol{z}$.

First-order characterization of convexity: $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}), \forall \boldsymbol{x}, \boldsymbol{y}$.

■

Thus, we always have a computable upper bound $g(\boldsymbol{x}_t)$ on the unknown error $f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)$, which is not the case in general for *unconstrained* optimization. Furthermore, at an optimal point $\boldsymbol{x}^\star$, $g(\boldsymbol{x}^\star) = 0$, which follows from the optimality conditions for constrained convex optimization.[7]

[7] $\nabla f(\boldsymbol{x})^\top (\boldsymbol{x} - \boldsymbol{x}^\star) \geq 0, \forall \boldsymbol{x} \in \mathcal{X}$.

7.3 *Convergence analysis*

To prove convergence, we need the following descent lemma,

Lemma 57. For a step $\boldsymbol{x}_{t+1} \doteq \boldsymbol{x}_t + \gamma_t(\boldsymbol{s} - \boldsymbol{x}_t)$ with stepsize $\gamma_t \in [0,1]$, it holds that

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \gamma_t g(\boldsymbol{x}_t) + \gamma_t^2 \frac{L}{2}\|\boldsymbol{s} - \boldsymbol{x}_t\|^2,$$

where $\boldsymbol{s} = \mathrm{LMO}_{\mathcal{X}}(\boldsymbol{\nabla} f(\boldsymbol{x}_t))$.

Proof.

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) &= f(\boldsymbol{x}_t + \gamma_t(\boldsymbol{s} - \boldsymbol{x}_t)) \\ &\leq f(\boldsymbol{x}_t) + \boldsymbol{\nabla} f(\boldsymbol{x}_t)^\top \gamma_t(\boldsymbol{s} - \boldsymbol{x}_t) + \gamma_t^2 \frac{L}{2}\|\boldsymbol{s} - \boldsymbol{x}_t\|^2 && \text{Smoothness:} \\ &= f(\boldsymbol{x}_t) - \gamma_t g(\boldsymbol{x}_t) + \gamma_t^2 \frac{L}{2}\|\boldsymbol{s} - \boldsymbol{x}_t\|^2. && \text{Duality gap: } g(\boldsymbol{x}) \doteq \boldsymbol{\nabla} f(\boldsymbol{x})^\top(\boldsymbol{x} - \boldsymbol{s}). \end{aligned}$$

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

■

Now, we can prove the main convergence theorem of the Frank-Wolfe algorithm.

Theorem 58. Consider the constrained minimization problem where $f : \mathbb{R}^d \to \mathbb{R}$ is convex and smooth with parameter L, and $\mathcal{X}$ is convex, closed and bounded. (This means that the minimizer $\boldsymbol{x}^\star$ of f over $\mathcal{X}$ exists and that all minimization oracles have minimizers.) With any $\boldsymbol{x}_0 \in \mathcal{X}$ and stepsizes

$$\gamma_t \doteq \frac{2}{t+2},$$

the Frank-Wolfe algorithm yields

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \leq \frac{2L}{T+1}\mathrm{diam}(\mathcal{X})^2,$$

where $\mathrm{diam}(\boldsymbol{x}) \doteq \max_{\boldsymbol{x},\boldsymbol{y} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|$ is the diameter of $\mathcal{X}$.

Proof. Let $h(\boldsymbol{x}) \doteq f(\boldsymbol{x}) - f(\boldsymbol{x}^\star)$ and $C \doteq \frac{L}{2}\mathrm{diam}(\mathcal{X})^2$.

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}^\star) &\leq f(\boldsymbol{x}_t) - \gamma_t g(\boldsymbol{x}_t) + \gamma_t^2 \frac{L}{2}\|\boldsymbol{s} - \boldsymbol{x}_t\|^2 - f(\boldsymbol{x}^\star) && \text{Lemma 57 and subtract } f(\boldsymbol{x}^\star) \text{ from both sides.} \\ h(\boldsymbol{x}_{t+1}) &\leq h(\boldsymbol{x}_t) - \gamma_t h(\boldsymbol{x}_t) + \gamma_t^2 \frac{L}{2}\|\boldsymbol{s} - \boldsymbol{x}_t\|^2 && \text{Duality gap, } g(\boldsymbol{x}) \geq h(\boldsymbol{x}). \\ &= (1 - \gamma_t)h(\boldsymbol{x}_t) + \gamma_t^2 \frac{L}{2}\|\boldsymbol{s} - \boldsymbol{x}_t\|^2 \\ &\leq (1 - \gamma_t)h(\boldsymbol{x}_t) + \gamma_t^2 C. \end{aligned}$$

Using our new definitions, we want to prove

$$h(\boldsymbol{x}_T) \leq \frac{4C}{T+1}, \quad T \geq 1.$$

We can prove this by induction. The base case is $T = 1$,

$$\begin{aligned} h(\boldsymbol{x}_1) &\leq (1-\gamma_0)h(\boldsymbol{x}_0) + \gamma_0^2 C \\ &= C && \gamma_0 = 1. \\ &\leq 2C. \end{aligned}$$

Suppose it holds for $T = k$,

$$h(\boldsymbol{x}_k) \leq \frac{4C}{k+1}.$$

then we need to show that it also holds for $T = k+1$,

$$\begin{aligned} h(\boldsymbol{x}_{k+1}) &\leq (1-\gamma_k)h(\boldsymbol{x}_k) + \gamma_k^2 C \\ &\leq \left(1 - \frac{2}{k+2}\right)\frac{4C}{k+1} + \left(\frac{2}{k+2}\right)^2 C && \text{Induction step and } \gamma_k = 2/k+2. \\ &= \frac{k}{k+2}\frac{4C}{k+1} + \frac{4}{(k+2)^2}C \\ &= \frac{4C}{k+2}\left(\frac{k}{k+1} + \frac{1}{k+2}\right) \\ &\leq \frac{4C}{k+2}. \end{aligned}$$

Thus, it holds for all $T \geq 1$. ■

Affine invariance. Consider the problem of minimizing $f(x_1, x_2) \doteq x_1^2 + x_2^2$ over the unit square $\mathcal{X} = \{[x_1, x_2] \mid 0 \leq x_1 \leq 1, 0 \leq x2 \leq 1\}$. This function is smooth with $L = 2$ and $\operatorname{diam}(\mathcal{X})^2 = 2$. This has the following error bound,

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \leq \frac{8}{T+1}.$$

Next consider $f'(x_1, x_2) \doteq x_1^2 + (10x_2)^2$ over the rectangle $\mathcal{X}'' = \{[x_1, x_2] \mid 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1/10\}$. This function is smooth with $L' = 200$ and $\operatorname{diam}(\mathcal{X}')^2 = 1 + 1/100$. Thus, f' has the error bound

$$f'(\boldsymbol{x}_T) - f'(\boldsymbol{x}^\star) \leq \frac{404}{T+1}.$$

Hence, according to our analysis, it seems that the error of the Frank-Wolfe algorithm on f' over $\mathcal{X}'$ is roughly 50 times larger than on f over $\mathcal{X}$.

However, when we look more closely at the function, the two problems $(f, \mathcal{X})$ and $(f', \mathcal{X}')$ are equivalent under a rescaling of x_2. The Frank-Wolfe algorithm is invariant under affine transformations, thus there should be no difference in the analysis of the two considered problems.

Formally, two problems $(f, \mathcal{X})$ and $(f', \mathcal{X}')$ are *affinely equivalent* if $f'(\boldsymbol{x}) = f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$ and $\mathcal{X}' = \{\boldsymbol{A}^{-1}(\boldsymbol{x} - \boldsymbol{b}) \mid \boldsymbol{x} \in \mathcal{X}\}$ for some invertible matrix $\boldsymbol{A}$ and some vector $\boldsymbol{b}$. The consequence is that $f(\boldsymbol{x}) = f'(\boldsymbol{x}')$ if $\boldsymbol{x}' = \boldsymbol{A}^{-1}(\boldsymbol{x} - \boldsymbol{b})$.[8]

[8] In our example problem, this means that we have

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, \quad \boldsymbol{b} = \boldsymbol{0}.$$

By the chain rule, we get

$$\nabla f'(x') = A^\top \nabla f(Ax' + b) = A^\top f(x).$$

Consider the iterate x_k and its corresponding iterate $x'_k = A^{-1}(x_k - b)$, in their respective problems. We can compute the their oracle calls by

$$\begin{aligned}
\mathrm{LMO}_{\mathcal{X}}(\nabla f(x_k)) &= \operatorname*{argmin}_{z \in \mathcal{X}} \nabla f(x_k)^\top z \\
&\doteq s \\
\mathrm{LMO}_{\mathcal{X}'}(\nabla f'(x'_k)) &= \operatorname*{argmin}_{z' \in \mathcal{X}'} \nabla f'(x'_k)^\top z' \\
&= \operatorname*{argmin}_{A^{-1}(z-b) \in \mathcal{X}'} \nabla f(x_k)^\top A A^{-1}(z - b) \\
&= \operatorname*{argmin}_{A^{-1}(z-b) \in \mathcal{X}'} \nabla f(x_k)^\top z - \nabla f(x_k)^\top b \\
&= A^{-1}\left(\left(\operatorname*{argmin}_{z \in \mathcal{X}} \nabla f(x_k)^\top z\right) - b\right) \\
&= A^{-1}(s - b).
\end{aligned}$$

Thus, the step directions s' and s also correspond to each other under the affine transformation. As a consequence, the next iterates will also correspond to each other,

$$\begin{aligned}
x_{k+1} &= (1-\gamma)x_k + \gamma s \\
x'_{k+1} &= (1-\gamma)x'_k + \gamma s' \\
&= (1-\gamma)A^{-1}(x_k - b) + \gamma A^{-1}(s - b) \\
&= A^{-1}((1-\gamma)x_k + \gamma s) - b \\
&= A^{-1}x_{k+1} - b.
\end{aligned}$$

In particular, after any number of steps, both problems will incur the same optimization error. Thus, we need a better analysis that provides a bound that is invariant under affine transformations. For this, we define a *curvature constant*,

$$C_{(f,\mathcal{X})} \doteq \sup_{\substack{x,s \in \mathcal{X}, \gamma \in (0,1] \\ y = (1-\gamma)x + \gamma s)}} \frac{1}{\gamma^2}\Big(f(y) - f(x) - \nabla f(x)^\top (y - x)\Big).$$

This quantity serves as a notion of complexity of both the objective function f and the constraint set $\mathcal{X}$. It is essentially the supremum of the normalized pointwise vertical distance between the graph of f, $f(y)$ and its linear approximation at x, $f(x) + \nabla f(x)^\top (y - x)$.

Theorem 59. Consider the constrained minimization problem, where $f : \mathbb{R}^d \to \mathbb{R}$ is convex, and $\mathcal{X}$ is convex, closed and bounded. Let $C_{(f,\mathcal{X})}$ be the curvature constant of f over $\mathcal{X}$. With any $\boldsymbol{x}_0 \in \mathcal{X}$ and with stepsizes

$$\gamma_t = \frac{2}{t+2},$$

the Frank-Wolfe algorithm yields

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^\star) \leq \frac{4C_{(f,\mathcal{X})}}{T+1}.$$

Proof. We can regain the descent lemma by rewriting the curvature constant. We know by the definition of the supremum,

$$\frac{1}{\gamma^2}\Big(f(\boldsymbol{y}) - f(\boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{x})^\top(\boldsymbol{y}-\boldsymbol{x})\Big) \leq C_{(f,\mathcal{X})}. \qquad \forall \boldsymbol{x}, \boldsymbol{s} \in \mathcal{X}, \gamma \in (0,1], \boldsymbol{y} = (1-\gamma)\boldsymbol{x} + \gamma \boldsymbol{s}.$$

Setting the variables,

$$\boldsymbol{x} \doteq \boldsymbol{x}_t, \quad \boldsymbol{y} \doteq \boldsymbol{x}_{t+1} = (1-\gamma_t)\boldsymbol{x}_t + \gamma_t \boldsymbol{s}, \quad \boldsymbol{y} - \boldsymbol{x} = -\gamma_t(\boldsymbol{x} - \boldsymbol{s}),$$

we get

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) &\leq f(\boldsymbol{x}_t) + \boldsymbol{\nabla} f(\boldsymbol{x}_t)^\top(-\gamma_t(\boldsymbol{x}-\boldsymbol{s})) + \gamma_t^2 C_{(f,\mathcal{X})} \\ &= f(\boldsymbol{x}_t) - \gamma g(\boldsymbol{x}_t) + \gamma_t^2 C_{(f,\mathcal{X})}. \end{aligned}$$

The rest of the proof follows as in the previous analysis. ■

You might suspect that this bound is worse than the best bound obtainable from the previous analysis. However, one can show

$$C_{(f,\mathcal{X})} \leq \frac{L}{2}\mathrm{diam}(\mathcal{X})^2.$$

Furthermore, we can prove a convergence of the duality gap.

Theorem 60. Let $f : \mathbb{R}^d \to \mathbb{R}$, $\mathcal{X}$ convex with $C_{(f,\mathcal{X})} < \infty$, $\boldsymbol{x}_0 \in \mathcal{X}$, and $T \geq 2$. Then, choosing stepsize

$$\gamma_t = \frac{2}{t+2},$$

the Frank-Wolfe algorithm yields a t with $1 \leq t \leq T$, such that

$$g(\boldsymbol{x}_t) \leq \frac{27/2 \cdot C_{(f,\mathcal{X})}}{T+1}.$$

8 Newton's method

The *Newton-Raphson method* is an iterative method for finding a zero of a differentiable univariate function $f : \mathbb{R} \to \mathbb{R}$. Starting from some x_0, it iteratively computes

$$x_{t+1} \doteq x_t - \frac{f(x_t)}{f'(x_t)}.$$

In formulas, x_{t+1} is the solution to the linear equation

$$f(x_t) + f'(x_t)(x - x_t) = 0,$$

yielding the above update formula. The Newton step fails if $f'(x_t) = 0$ or gets out of control if $|f'(x_t)|$ is very small. Thus, we need to keep this in mind when making a theoretical analysis.

We can use this method for optimization as well, called *Newton's method*, where we can find critical points $f'(x) = 0$ by applying the method to the derivative of f,

$$x_{t+1} \doteq x_t - \frac{f'(x_t)}{f''(x_t)}.$$

We can further generalize this to any dimensionality,

$$\boldsymbol{x}_{t+1} \doteq \boldsymbol{x}_t - \boldsymbol{\nabla}^2 f(\boldsymbol{x}_t)^{-1} \boldsymbol{\nabla} f(\boldsymbol{x}_t).$$

As before, we need to keep in mind that the Hessian must be invertible and may get out of control if the Hessian has small norm.

A second interpretation of Newton's method is that it is a special case of the general update scheme,

$$\boldsymbol{x}_{t+1} \doteq \boldsymbol{x}_t - H(\boldsymbol{x}_t) \boldsymbol{\nabla} f(\boldsymbol{x}_t),$$

where $H(\boldsymbol{x}_t) \in \mathbb{R}^{d \times d}$ is some matrix, like gradient descent with $H(\boldsymbol{x}_t) = \gamma_t \boldsymbol{I}$. Hence, we can think of Newton's method of an "adaptive gradient descent" that adapts to the local curvature of the function at $\boldsymbol{x}_t$.[9]

We will not prove any general convergence guarantees for Newton's method. We will prove that, under suitable conditions, and starting close to a critical point, we will reach distance at most ϵ to this critical point in $\mathcal{O}(\log\log(1/\epsilon))$ steps. This also holds for non-convex functions. However, this is quite weak, since we assume that we are already close to the critical point. The proof will rely on the assumption that the local curvature in the small space around the critical point is near constant.

The *Babylonian method* to compute square roots is an application of the Newton-Raphson method. It finds zeros of $f(x) = x^2 - R$, which is equal to zero at $\sqrt{R}$ and $-\sqrt{R}$. It takes $\mathcal{O}(\log R)$ steps to get within $1/2$ of a square root. Then, to get within ϵ, it takes $\log\log(1/\epsilon)$ steps. Thus, once we are close, we get very close very quickly.

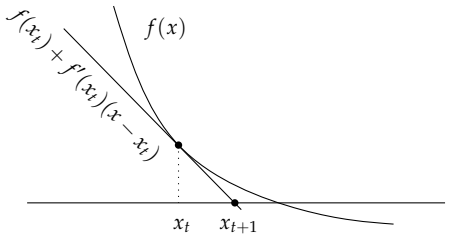


Figure 22. A step of the Newton-Raphson method.

[9] This is very apparent in the case of optimizing a quadratic function of the form $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x} - \boldsymbol{q}^\top \boldsymbol{x} + c$, which has the same curvature $\boldsymbol{\nabla}^2 f(\boldsymbol{x}) = \boldsymbol{M}$ everywhere. In this case, Newton's method yields the optimum in a single step, $\boldsymbol{x}_1 = \boldsymbol{x}^\star$.

Theorem 61. Let $f : \mathrm{dom}(f) \to \mathbb{R}$ be twice differentiable with a critical point $\boldsymbol{x}^\star$. Suppose there is a ball $\mathcal{X} \subseteq \mathrm{dom}(f)$ with center $\boldsymbol{x}^\star$ such that the inverse Hessians are bounded,

$$\exists \mu > 0, \quad \left\| \boldsymbol{\nabla}^2 f(\boldsymbol{x})^{-1} \right\| \leq \frac{1}{\mu}, \quad \forall \boldsymbol{x} \in \mathcal{X},$$

and the Hessian is Lipschitz continuous,

$$\exists B \geq 0, \quad \left\| \boldsymbol{\nabla}^2 f(\boldsymbol{x}) - \boldsymbol{\nabla}^2 f(\boldsymbol{y}) \right\| \leq B \|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}.$$

Then, for $\boldsymbol{x}_t \in \mathcal{X}$ and $\boldsymbol{x}_{t+1}$, resulting from the Newton step, we have

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\| \leq \frac{B}{2\mu} \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2.$$

Proof. Let $H(\boldsymbol{x}) = \boldsymbol{\nabla}^2 f(\boldsymbol{x})$, $\boldsymbol{x} = \boldsymbol{x}_t$, $\boldsymbol{x}' = \boldsymbol{x}_{t+1}$. Then, subtracting $\boldsymbol{x}^\star$ from both sides of the Newton step yields

$$\begin{aligned}
\boldsymbol{x}' - \boldsymbol{x}^\star &= \boldsymbol{x} - \boldsymbol{x}^\star - H(\boldsymbol{x})^{-1} \boldsymbol{\nabla} f(\boldsymbol{x}) \\
&= \boldsymbol{x} - \boldsymbol{x}^\star + H(\boldsymbol{x})^{-1} (\boldsymbol{\nabla} f(\boldsymbol{x}^\star) - \boldsymbol{\nabla} f(\boldsymbol{x})). && \boldsymbol{\nabla} f(\boldsymbol{x}^\star) = \boldsymbol{0}.
\end{aligned}$$

Let $h(t) \doteq \boldsymbol{\nabla} f(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x}))$. Then, using the fundamental theorem of calculus, we get

$$= \boldsymbol{x} - \boldsymbol{x}^\star + H(\boldsymbol{x})^{-1} \int_0^1 H(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x}))(\boldsymbol{x}^\star - \boldsymbol{x}) \mathrm{d}t \qquad h'(t) = H(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x}))(\boldsymbol{x}^\star - \boldsymbol{x}).$$

Using $\boldsymbol{x} - \boldsymbol{x}^\star = H(\boldsymbol{x})^{-1} H(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{x}^\star) = H(\boldsymbol{x})^{-1} \int_0^1 -H(\boldsymbol{x})(\boldsymbol{x}^\star - \boldsymbol{x}) \mathrm{d}t$, we get

$$= H(\boldsymbol{x})^{-1} \int_0^1 (H(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x})) - H(\boldsymbol{x}))(\boldsymbol{x}^\star - \boldsymbol{x}) \mathrm{d}t$$

Taking norm of both sides yields

$$\begin{aligned}
\|\boldsymbol{x}' - \boldsymbol{x}\| &= \left\| H(\boldsymbol{x})^{-1} \int_0^1 (H(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x})) - H(\boldsymbol{x}))(\boldsymbol{x}^\star - \boldsymbol{x}) \mathrm{d}t \right\| \\
&\leq \left\| H(\boldsymbol{x})^{-1} \right\| \cdot \left\| \int_0^1 (H(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x})) - H(\boldsymbol{x}))(\boldsymbol{x}^\star - \boldsymbol{x}) \mathrm{d}t \right\| && \|\boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{A}\| \cdot \|\boldsymbol{x}\|. \\
&\leq \left\| H(\boldsymbol{x})^{-1} \right\| \cdot \int_0^1 \| (H(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x})) - H(\boldsymbol{x}))(\boldsymbol{x}^\star - \boldsymbol{x}) \| \mathrm{d}t \\
&\leq \left\| H(\boldsymbol{x})^{-1} \right\| \cdot \int_0^1 \| H(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x})) - H(\boldsymbol{x}) \| \cdot \|\boldsymbol{x}^\star - \boldsymbol{x}\| \mathrm{d}t && \|\boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{A}\| \cdot \|\boldsymbol{x}\|. \\
&\leq \left\| H(\boldsymbol{x})^{-1} \right\| \cdot \|\boldsymbol{x}^\star - \boldsymbol{x}\| \int_0^1 \| H(\boldsymbol{x} + t(\boldsymbol{x}^\star - \boldsymbol{x})) - H(\boldsymbol{x}) \| \mathrm{d}t \\
&\leq \frac{1}{\mu} \|\boldsymbol{x}^\star - \boldsymbol{x}\| \int_0^1 B \| t(\boldsymbol{x}^\star - \boldsymbol{x}) \| \mathrm{d}t && \text{Assumptions of theorem.} \\
&\leq \frac{B}{\mu} \|\boldsymbol{x}^\star - \boldsymbol{x}\|^2 \int_0^1 t \mathrm{d}t \\
&\leq \frac{B}{2\mu} \|\boldsymbol{x}^\star - \boldsymbol{x}\|^2.
\end{aligned}$$

■

Corollary. With the assumptions of Theorem 61, if $x_0 \in \mathcal{X}$ satisfies

$$\|x_0 - x^\star\| \leq \frac{\mu}{B},$$

then Newton's method yields

$$\|x_T - x^\star\| \leq \frac{\mu}{B}\left(\frac{1}{2}\right)^{2^T-1}.$$

Hence, we get the $\mathcal{O}(\log\log(1/\epsilon))$ bound, but only if we are μ/B-close to $x^\star$. Thus, we only converge fast to $x^\star$ if we are already close to it. For this to hold, it is of course necessary that $x^\star$ is the *only* close critical point to x_0. However, this necessarily follows from the assumptions, since the Hessians are almost constant this close to $x^\star$ under the Lipschitz continuity and inverse Hessian bound. Thus, locally, the function behaves like a quadratic function, which converges to its unique critical point in one step.

9 *Quasi-Newton methods*

The problem with Newton's method is that it has a high computational complexity, due to the Hessian and inverse of it, which both have $\mathcal{O}(d^3)$ runtime complexity. *Quasi-Newton methods* are optimization methods that approximate the Hessian by a matrix $\boldsymbol{H}_t \approx \boldsymbol{\nabla}^2 f(\boldsymbol{x}_t)$, which is a function of $\boldsymbol{x}_t$, $\boldsymbol{x}_{t-1}$, and $\boldsymbol{H}_{t-1}$. We then iteratively update by

$$\boldsymbol{x}_{t+1} \doteq \boldsymbol{x}_t - \boldsymbol{H}_t^{-1} \boldsymbol{\nabla} f(\boldsymbol{x}_t),$$

where $\boldsymbol{H}_t \in \mathbb{R}^{d \times d}$ must be symmetric and satisfy the *secant condition*,

$$\boldsymbol{\nabla} f(\boldsymbol{x}_t) - \boldsymbol{\nabla} f(\boldsymbol{x}_{t-1}) = \boldsymbol{H}_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}).$$

In general, there are many matrices that satisfy these conditions. Thus, we must choose which $\boldsymbol{H}_t^{-1}$ to pick, based on $\boldsymbol{x}_{t-1}$, $\boldsymbol{x}_t$, and $\boldsymbol{H}_t^{-1}$.[10]

[10] We directly work with $\boldsymbol{H}_t^{-1}$, instead of $\boldsymbol{H}_t$, since computing the inverse would again result in a $\mathcal{O}(d^3)$ runtime complexity.

Recall from Newton's method that we wanted $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_t)$ to fluctuate very little in regions of fast convergence. Hence, in Quasi-Newton methods, it makes sense if $\boldsymbol{H}_{t-1}^{-1} \approx \boldsymbol{H}_t^{-1}$. This intuition yields the approach by Greenstadt [1970], where we update $\boldsymbol{H}_{t-1}^{-1}$ by an error matrix $\boldsymbol{E}_t$,

$$\boldsymbol{H}_t^{-1} = \boldsymbol{H}_{t-1}^{-1} + \boldsymbol{E}_t,$$

and we want this error to be as small as possible, *i.e.* minimize $\|\boldsymbol{E}\|_F^2$, subject to its constraints. Greenstadt [1970] found this method "too specialized", which lead him to minimize the following error term instead,

$$\|\boldsymbol{A}\boldsymbol{E}\boldsymbol{A}^\top\|_F^2,$$

where $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ is a fixed invertible transformation matrix.

Let's now use the following notation to develop further algorithms,

$$\begin{aligned}
\boldsymbol{H} &\doteq \boldsymbol{H}_{t-1}^{-1} \\
\boldsymbol{H}' &\doteq \boldsymbol{H}_t^{-1} \\
\boldsymbol{E} &\doteq \boldsymbol{E}_t \\
\boldsymbol{\sigma} &\doteq \boldsymbol{x}_t - \boldsymbol{x}_{t-1} \\
\boldsymbol{y} &\doteq \boldsymbol{\nabla} f(\boldsymbol{x}_t) - \boldsymbol{\nabla} f(\boldsymbol{x}_{t-1}) \\
\boldsymbol{r} &\doteq \boldsymbol{\sigma} - \boldsymbol{H}\boldsymbol{y}.
\end{aligned}$$

We then have the following convex constrained minimization problem in d^2 variables,

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{E}\boldsymbol{A}^\top\|_F^2 \\
\text{subject to} \quad & \boldsymbol{E}\boldsymbol{y} = \boldsymbol{r} \\
& \boldsymbol{E}^\top - \boldsymbol{E} = \boldsymbol{0},
\end{aligned}$$

where the first condition is the secant condition,

$$\boldsymbol{H}'\boldsymbol{y} = \boldsymbol{\sigma} \Leftrightarrow (\boldsymbol{H} + \boldsymbol{E})\boldsymbol{y} = \boldsymbol{\sigma} \Leftrightarrow \boldsymbol{E}\boldsymbol{y} = \boldsymbol{\sigma} - \boldsymbol{H}\boldsymbol{y} \Leftrightarrow \boldsymbol{E}\boldsymbol{y} = \boldsymbol{r},$$

and the second condition ensures symmetry, since if $\boldsymbol{H}_{t-1}^{-1}$ and $\boldsymbol{E}_t$ are symmetric, then $\boldsymbol{H}_t^{-1}$ is as well.

Let

$$f(\boldsymbol{E}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{E}\boldsymbol{A}^\top\|_F^2.$$

Because the conditions are all linear, we can summarize them in one equation as $\boldsymbol{C}\boldsymbol{E} = \boldsymbol{B}$ for some matrices $\boldsymbol{C}$ and $\boldsymbol{B}$. Furthermore, due to this convex program only having equality constraints, the Slater point condition for strong duality becomes void. Thus, we obtain strong duality "for free". Thus, the Karush-Kuhn-Tucker conditions hold, which imply there exists a vector $\boldsymbol{\lambda} \in \mathbb{R}^m$ such that

$$\boldsymbol{\nabla} f(\boldsymbol{E}^\star)^\top = \boldsymbol{\lambda}^\top \boldsymbol{C}.$$

Directly follows from the vanishing gradient condition in KTT.

Let $\boldsymbol{W} = \boldsymbol{A}^\top \boldsymbol{A}$ and $\boldsymbol{M} = \boldsymbol{W}^{-1}$, then the gradient of f can be computed by

$$\boldsymbol{\nabla} f(\boldsymbol{E}) = \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{E}\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{W}\boldsymbol{E}\boldsymbol{W} = \boldsymbol{M}^{-1}\boldsymbol{E}\boldsymbol{M}^{-1}.$$

Now, since the objective is quadratic, we can obtain the minimizer $\boldsymbol{x}^\star$ and the Lagrange multipliers $\boldsymbol{\lambda}$ by solving the following system of linear equations,

$$\begin{aligned} \boldsymbol{C}\boldsymbol{E} &= \boldsymbol{B} \\ \boldsymbol{E} &= \boldsymbol{M}^\top \boldsymbol{\lambda}^\top \boldsymbol{C}\boldsymbol{M}^\top. \end{aligned}$$

Solving this system yields

$$\begin{aligned} \boldsymbol{E}^\star = \frac{1}{\boldsymbol{y}^\top \boldsymbol{M}\boldsymbol{y}}\Big(&\sigma \boldsymbol{y}^\top \boldsymbol{M} + \boldsymbol{M}\boldsymbol{y}\sigma^\top - \boldsymbol{H}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{M} - \boldsymbol{M}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{H} \\ &- \frac{1}{\boldsymbol{y}^\top \boldsymbol{M}\boldsymbol{y}}\Big(\boldsymbol{y}^\top \sigma - \boldsymbol{y}^\top \boldsymbol{H}\boldsymbol{y}\Big)\boldsymbol{M}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{M}\Big). \end{aligned}$$

This is called the *Greenstadt method* with parameter $\boldsymbol{M}$.

Now, we need to decide which $\boldsymbol{M}$ to use. Greenstadt [1970] suggested $\boldsymbol{M} = \boldsymbol{I}$ and $\boldsymbol{M} = \boldsymbol{H} \doteq \boldsymbol{H}_{t-1}^{-1}$. Goldfarb [1970] suggested $\boldsymbol{M} = \boldsymbol{H}' \doteq \boldsymbol{H}_t^{-1}$. Because of the secant condition, we get the following,

$$\boldsymbol{M}\boldsymbol{y} = \boldsymbol{H}'\boldsymbol{y} = \sigma.$$

Hence, despite not knowing this value yet, we can still use it, since it will cancel out all terms, containing $\boldsymbol{M} = \boldsymbol{H}'$. This is called the *BFGS method*, and the optimal error matrix becomes

$$\boldsymbol{E}^\star = \frac{1}{\boldsymbol{y}^\top \sigma}\left(-\boldsymbol{H}\boldsymbol{y}\sigma^\top - \sigma \boldsymbol{y}^\top \boldsymbol{H} + \left(1 + \frac{\boldsymbol{y}^\top \boldsymbol{H}\boldsymbol{y}}{\boldsymbol{y}^\top \sigma}\right)\sigma\sigma^\top\right).$$

With this error matrix, we get the following update,

$$\boldsymbol{H}' = \left(\boldsymbol{I} - \frac{\sigma \boldsymbol{y}^\top}{\boldsymbol{y}^\top \sigma}\right)\boldsymbol{H}\left(\boldsymbol{I} - \frac{\boldsymbol{y}\sigma^\top}{\boldsymbol{y}^\top \sigma}\right) + \frac{\sigma\sigma^\top}{\boldsymbol{y}^\top \sigma}.$$

The cost per step of this algorithm is $\mathcal{O}(d^2)$, which is a big upgrade over $\mathcal{O}(d^3)$ that we had for Newton's method. However, we can make it even faster by making another approximation, which will yield the *L-BFGS* algorithm.

Recall the Quasi-Newton update step,

$$\boldsymbol{x}_{t+1} \doteq \boldsymbol{x}_t - \boldsymbol{H}_t^{-1} \boldsymbol{\nabla} f(\boldsymbol{x}_t).$$

Observe that we do not necessarily need the $d \times d$ matrix $\boldsymbol{H}_t^{-1}$; we only need the d-dimensional vector $\boldsymbol{H}_t^{-1} \boldsymbol{\nabla} f(\boldsymbol{x}_t)$. Let $\boldsymbol{g}' \in \mathbb{R}^d$. Suppose that we have an oracle to compute $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{g}$ for any vector $\boldsymbol{g}$, then we can compute $\boldsymbol{s}' = \boldsymbol{H}'\boldsymbol{g}'$ with one oracle call and $\mathcal{O}(d)$ additional operations, assuming that $\boldsymbol{y}$ and $\boldsymbol{\sigma}$ are known.

We can implement the oracle recursively,

$$\begin{aligned} \boldsymbol{\sigma}_k &\doteq \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \\ \boldsymbol{y}_k &\doteq \boldsymbol{\nabla} f(\boldsymbol{x}_k) - \boldsymbol{\nabla} f(\boldsymbol{x}_{k+1}). \end{aligned}$$

This allows us to compute the BFGS-step $\boldsymbol{H}_t^{-1} \boldsymbol{\nabla} f(\boldsymbol{x}_t)$ recursively. However, this would result in $\mathcal{O}(td)$ runtime complexity per step, since we would have to go down all steps, and generally $t > d$. Thus, we have a worse algorithm if we want to compute the next vector exactly. But, if we only go down m steps of recursion for some small m, we get $\mathcal{O}(md)$ complexity, which is linear if m is constant; see Algorithm 1. Intuitively, this should give a good approximation, since the earlier steps should not be so relevant anymore, since we are likely in a different landscape at the current timestep.

function LBFGSSTEP($k, \ell, \boldsymbol{g}'$)
  if $\ell = 0$ **then**
    return $\boldsymbol{H}_0^{-1}\boldsymbol{g}'$
  end if
  $\boldsymbol{h} = \boldsymbol{\sigma} \frac{\boldsymbol{\sigma}_k^\top \boldsymbol{g}'}{\boldsymbol{y}_k^\top \boldsymbol{\sigma}_k}$
  $\boldsymbol{g} = \boldsymbol{g}' - \boldsymbol{y} \frac{\boldsymbol{\sigma}_k^\top \boldsymbol{g}'}{\boldsymbol{y}_k^\top \boldsymbol{\sigma}_k}$
  $\boldsymbol{s} = \text{LBFGSSTEP}(k-1, \ell-1, \boldsymbol{g})$
  $\boldsymbol{w} = \boldsymbol{s} - \boldsymbol{\sigma}_k \frac{\boldsymbol{y}_k^\top \boldsymbol{s}}{\boldsymbol{y}_k^\top \boldsymbol{\sigma}_k}$
  $\boldsymbol{z} = \boldsymbol{w} + \boldsymbol{h}$
  return $\boldsymbol{z}$
end function

Algorithm 1. The L-BFGS algorithm. The outer products can be computed as inner products, giving $\mathcal{O}(d)$ runtime complexity to all the products.

10 Subgradient methods

Until now, we have mostly assumed all functions to be smooth and hence differentiable. However, in general this is not the case. In machine learning, non-smoothness arises everywhere:

- Loss functions, such as the Hinge loss, $\max\{0, 1-x\}$ (SVM);
- Regularization, such as the ℓ_1-norm (LASSO);
- Activation functions, such as ReLU.

This motivates the need for a more general notion of the gradient that can be applied to more functions.

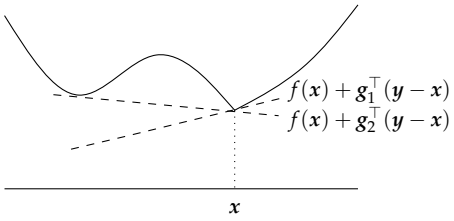


Figure 23. g is a subgradient of f at x if the whole graph is above x's supporting hyperplane, parametrized by g.

Definition 62 (Subgradient)**.** $g \in \mathbb{R}^d$ is a subgradient of f at x if

$$f(y) \geq f(x) + g^\top (y - x), \quad \forall y \in \mathrm{dom}(f).$$

We call $\partial f(x) \subseteq \mathbb{R}^d$ the subdifferential, which is the set of subgradients of f at x.

Example 63. Consider $f(x) = |x|$, then $\partial f(0) = [-1, 1]$.

Lemma 64. If f is differentiable at $x \in \mathrm{dom}(f)$, then $\partial f(x) \subseteq \{\nabla f(x)\}$.

Lemma 64 means that if f is differentiable at x, then this is either the only subgradient or there is no subgradient at all. There might not be any subgradient at all in this case because it might be that the hyperplane is not below the entire function if the function is non-convex.

Lemma 65. The following characterizes convexity with the subgradient:

- If f is convex, then $\partial f(x) \neq \emptyset$ for all x in the relative interior of $\mathrm{dom}(f)$;
- If $\mathrm{dom}(f)$ is convex and $\partial f(x) \neq \emptyset$ for all $x \in \mathrm{dom}(f)$, then f is convex.

Lemma 66 (Subgradient optimality condition)**.** Let $f : \mathrm{dom}(f) \to \mathbb{R}$ and $x \in \mathrm{dom}(f)$. If $\mathbf{0} \in \partial f(x)$, then x is a global minimum.

Proof. By definition of the subgradient with $g = \mathbf{0} \in \partial f(x)$ gives

$$f(y) \geq f(x) + g^\top (y - x) = f(x), \quad \forall y \in \mathrm{dom}(f).$$

Thus, x is a global minimum. ∎

Lemma 67 (Subgradient calculus)**.** We can use the following operations to work with subgradients:

- Conic combination: Let $h(\boldsymbol{x}) = \alpha f(\boldsymbol{x}) + \beta g(\boldsymbol{x})$ with $\alpha, \beta \geq 0$, then
$$\partial h(\boldsymbol{x}) = \alpha \partial f(\boldsymbol{x}) + \beta \partial g(\boldsymbol{x});$$
- Affine transformation: Let $h(\boldsymbol{x}) = f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$, then
$$\partial h(\boldsymbol{x}) = \boldsymbol{A}^\top \partial f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b});$$
- Pointwise maximum: Let $h(\boldsymbol{x}) = \max_{i \in [m]} f_i(\boldsymbol{x})$, then
$$\partial h(\boldsymbol{x}) = \operatorname{conv}(\{\partial f_i(\boldsymbol{x}) \mid f_i(\boldsymbol{x}) = h(\boldsymbol{x})\}).$$
Thus, at each point where we transition from one function to another, we get the convex hull of subgradients of the functions that transition. At all other points, we take the maximum function's subgradient.

10.1 Subgradient method

In the subgradient method, the general update rule becomes

$$\boldsymbol{x}_{t+1} = \Pi_{\mathcal{X}}(\boldsymbol{x}_t - \gamma_t \boldsymbol{g}_t), \quad \boldsymbol{g}_t \in \partial f(\boldsymbol{x}_t),$$

which can be rewritten as an optimization problem,

$$\begin{aligned}
&= \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{x}_t - \gamma_t \boldsymbol{g}_t)\|^2 \\
&= \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}_t - (-\gamma_t \boldsymbol{g}_t)\|^2 \\
&= \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{2}\Big(\|\boldsymbol{x} - \boldsymbol{x}_t\|^2 + \|\gamma_t \boldsymbol{g}_t\|^2 + 2\langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x} - \boldsymbol{x}_t\rangle\Big) && \text{Cosine theorem.} \\
&= \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{X}} \left\{\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}_t\|^2 + \langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x}\rangle\right\}. && \text{Remove terms that do not depend on } \boldsymbol{x}.
\end{aligned}$$

If f is differentiable, gradient descent and projected gradient descent are special cases of this update rule, where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{X} \subset \mathbb{R}^d$, respectively. However, if f is non-differentiable, we will see that this is technically not a descent method, because the subgradient is not a descent direction in general.

Lemma 68 (Subgradient method "descent" lemma)**.** If f is convex, then for any optimal solution $\boldsymbol{x}^\star$,

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\gamma_t(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) + \gamma_t^2 \|\boldsymbol{g}_t\|^2.$$

Proof.

$$\begin{aligned}
\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 &= \|\Pi_{\mathcal{X}}(\boldsymbol{x}_t - \gamma_t \boldsymbol{g}_t) - \boldsymbol{x}^\star\|^2 && \text{Subgradient descent update rule.} \\
&= \|\boldsymbol{x}_t - \gamma_t \boldsymbol{g}_t - \boldsymbol{x}^\star\|^2 && \mathcal{X} \text{ is convex, so we get closer to } \boldsymbol{x}^\star \text{ after projection.} \\
&= \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\gamma_t \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^\star) + \gamma_t^2 \|\boldsymbol{g}_t\|^2 && \text{Cosine theorem: } \|\boldsymbol{x} - \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{x}^\top \boldsymbol{y}. \\
&\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\gamma_t (f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) + \gamma_t^2 \|\boldsymbol{g}_t\|^2. && \text{Convexity: } f(\boldsymbol{x}^\star) \geq f(\boldsymbol{x}_t) + \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^\star).
\end{aligned}$$

■

Theorem 69 (Convergence of the subgradient method)**.** If f is convex, then the subgradient method satisfies

$$\min_{t \in [T]} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sum_{t=0}^{T-1} \gamma_t^2 \|\boldsymbol{g}_t\|^2}{2 \sum_{t=0}^{T-1} \gamma_t}.$$

Proof. This can easily be shown by telescoping sum of Lemma 68 and invoking convexity. ■

Assuming bounded subgradient $\|\boldsymbol{g}_t\| \leq B$ for all steps t, we get the following convergence rates under various stepsizes,

- Constant stepsize ($\gamma_t = \gamma$):

$$\lim_{t \to \infty} f(\boldsymbol{x}_t^{\text{best}}) \leq f(\boldsymbol{x}^\star) + \frac{B^2 \gamma}{2};$$

- Scaled stepsize ($\gamma_t = \gamma / \|\boldsymbol{g}_t\|$):

$$\lim_{t \to \infty} f(\boldsymbol{x}_t^{\text{best}}) \leq f(\boldsymbol{x}^\star) + \frac{B \gamma}{2};$$

- Square-summable stepsize ($\sum_{t=0}^{\infty} \gamma_t^2 < +\infty$, $\sum_{t=0}^{\infty} \gamma_t = +\infty$):

$$\lim_{t \to \infty} f(\boldsymbol{x}_t^{\text{best}}) = f(\boldsymbol{x}^\star);$$

- Diminishing stepsize ($\gamma_t \to 0$ and $\sum_{t=0}^{\infty} \gamma_t = +\infty$):

$$\lim_{t \to \infty} f(\boldsymbol{x}_t^{\text{best}}) = f(\boldsymbol{x}^\star).$$

Corollary. Let f be convex and B-Lipschitz continuous. Let $\mathcal{X}$ be convex compact with $R^2 = \max_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|_2 < +\infty$. Setting

$$\gamma \doteq \frac{R}{B\sqrt{T}},$$

then the subgradient method satisfies

$$\min_{t \in [T]} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq \frac{BR}{\sqrt{T}}.$$

To achieve ϵ-optimality, the subgradient method requires $\mathcal{O}\left(\frac{B^2R^2}{\epsilon^2}\right)$ iterations.

10.2 Strong convexity

Theorem 70. Let f be μ-strongly convex and B-Lipschitz continuous on $\mathcal{X}$. Setting

$$\gamma_t \doteq \frac{2}{\mu(t+1)},$$

then the subgradient method satisfies

$$\min_{t\in[T]} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq \frac{2B^2}{\mu(T+1)}.$$

In the case of strong convexity, to achieve ϵ-optimality, the subgradient method requires $\mathcal{O}\left(\frac{B^2}{\mu\epsilon}\right)$ iterations.

Proof. Adapting the proof of Lemma 68 to use strong convexity in its last step, we get

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \leq (1-\mu\gamma_t)\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\gamma_t(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) + \gamma_t^2\|\boldsymbol{g}_t\|^2.$$

Using this, we get the following,

$$\begin{aligned} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) &\leq \frac{1-\mu\gamma_t}{2\gamma_t}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \frac{1}{2\gamma_t}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 + \frac{\gamma_t}{2}\|\boldsymbol{g}_t\|^2 \\ &= \frac{\mu(t-1)}{4}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \frac{\mu(t+1)}{4}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 + \frac{1}{\mu(t+1)}\|\boldsymbol{g}_t\|^2 \cdot \gamma_t \doteq 2/\mu(t+1) \end{aligned}$$

Now, it is easy to show the result by a telescoping sum. ∎

11 Mirror descent

Like in subgradient descent, we continue to assume that f is non-smooth. In practice, we often have additional information about set $\mathcal{X}$ that we might be able to exploit. Specifically, we will explore how we can exploit non-Euclidean geometry of a convex set $\mathcal{X}$.[11]

[11] Until this point, we have only made use of Euclidean geometry by way of using the $\|\cdot\|_2$-norm.

11.1 Norm and Bregman divergence

Definition 71 (Norm)**.** A function $\|\cdot\| : \mathcal{X} \to \mathbb{R}_+$ is a norm if it satisfies the following properties,

1. (Positive definiteness) $\|\boldsymbol{x}\| = 0$ if and only if $\boldsymbol{x} = \boldsymbol{0}$;
2. (Positive homogeneity) $\|\alpha\boldsymbol{x}\| = |\alpha|\|\boldsymbol{x}\|$;
3. (Subadditivity): $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$.

Definition 72 (Dual norm)**.** The dual norm $\|\cdot\|_*$ of a norm $\|\cdot\|$ satisfies the properties of a norm and

$$\|\boldsymbol{y}\|_* \doteq \max_{\|\boldsymbol{x}\| \leq 1} \langle \boldsymbol{x}, \boldsymbol{y} \rangle.$$

Example 73. For $p \geq 1$ and $1/p + 1/q = 1$, we have the following norms with their dual norms,

$$\|\boldsymbol{x}\|_p \doteq \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p}, \quad \|\cdot\|_{p,*} = \|\cdot\|_q.$$

Lemma 74.

$$\frac{1}{\sqrt{d}} \|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_1 \leq \sqrt{d}\|\boldsymbol{x}\|_2.$$

The nice thing about smoothness, Lipschitz continuity, and strong convexity is that they can be defined for any norm.

Definition 75 (Bregman divergence)**.** Let $\omega : \Omega \to \mathbb{R}$ be continuously differentiable on Ω and 1-strongly convex w.r.t. some norm $\|\cdot\|$,

$$\omega(\boldsymbol{x}) \geq \omega(\boldsymbol{y}) + \nabla\omega(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega.$$

The Bregman divergence V_ω is defined as

$$V_\omega(\boldsymbol{x}, \boldsymbol{y}) \doteq \omega(\boldsymbol{x}) - \omega(\boldsymbol{y}) - \nabla\omega(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega.$$

Note that the Bregman divergence V_ω is thus the first-order Taylor approximation of ω, evaluated at $\boldsymbol{x}$.

Example 76. We have the following examples of Bregman divergences,

1. (Euclidean distance) $\Omega = \mathbb{R}^d$, $\omega(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$, and $\|\cdot\| = \|\cdot\|_2$. Then
$$V_\omega(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$
2. (Mahalanobis distance) $\Omega = \mathbb{R}^d$, $\omega(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$ with $\boldsymbol{Q} \succeq \boldsymbol{I}$, and $\|\cdot\| = \|\cdot\|_2$. Then,
$$V_\omega(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{y})^\top \boldsymbol{Q}(\boldsymbol{x} - \boldsymbol{y}).$$
3. (Kullback-Leibler divergence) $\Omega = \Delta^{d-1}$, $\omega(\boldsymbol{x}) = \sum_{i=1}^d x_i \log x_i$, and $\|\cdot\| = \|\cdot\|_1$. Then,
$$V_\omega(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{KL}(\boldsymbol{x}; \boldsymbol{y}) \doteq \sum_{i=1}^d x_i \log \frac{x_i}{y_i}.$$

Lemma 77. Any Bregman divergence satisfies the following properties:

1. (Non-negativity) $V_\omega(\boldsymbol{x}, \boldsymbol{y}) \geq 0$;
2. (Convexity) $V_\omega(\boldsymbol{x}, \boldsymbol{y})$ is convex in $\boldsymbol{x}$;
3. (Positivity) $V_\omega(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$;
4. $V_\omega(\boldsymbol{x}, \boldsymbol{y}) \geq \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$.

The following Lemma is a key property of the Bregman divergence and is used extensively in this course.

Lemma 78 (Three-point identity)**.** $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \Omega$:

$$V_\omega(\boldsymbol{x}, \boldsymbol{z}) = V_\omega(\boldsymbol{x}, \boldsymbol{y}) + V_\omega(\boldsymbol{y}, \boldsymbol{z}) - \langle \boldsymbol{\nabla}\omega(\boldsymbol{z}) - \boldsymbol{\nabla}\omega(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\rangle.$$

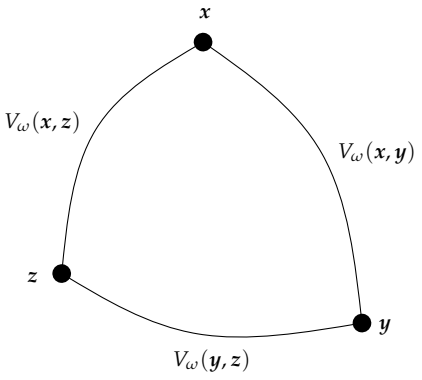


Figure 24. Illustration of the three-point identity of a non-Euclidean Bregman divergence.

11.2 *Mirror descent algorithm*

The mirror descent algorithm is a generalization of the subgradient method, where the norm is replaced by a Bregman divergence,[12]

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{X}} \{V_\omega(\boldsymbol{x}, \boldsymbol{x}_t) + \langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x}\rangle\}, \quad \boldsymbol{g}_t \in \partial f(\boldsymbol{x}_t).$$

[12] Using Item 1 of Example 76 recovers the subgradient method, and Item 3 recovers Entropic descent,

$$\boldsymbol{x}_{t+1} \propto \boldsymbol{x}_t \odot \exp(-\gamma_t \boldsymbol{g}_t).$$

Lemma 79. Let f be convex and ω be 1-strongly convex on $\mathcal{X}$ w.r.t. norm $\|\cdot\|$. Then, the following inequality holds,

$$\gamma_t(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) \leq V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_t) - V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_{t+1}) + \frac{\gamma_t^2}{2}\|\boldsymbol{g}_t\|_*^2.$$

Proof. We have the following update rule,

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{X}}\{V_\omega(\boldsymbol{x}, \boldsymbol{x}_t) + \langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x}\rangle\}.$$

Thus, by the optimality condition, we have

$$\langle \boldsymbol{\nabla}\omega(\boldsymbol{x}_{t+1}) + \gamma_t \boldsymbol{g}_t - \boldsymbol{\nabla}\omega(\boldsymbol{x}_t), \boldsymbol{x} - \boldsymbol{x}_{t+1}\rangle \geq 0, \quad \forall \boldsymbol{x} \in \mathcal{X},$$

which can be equivalently written as $\forall \boldsymbol{x} \in \mathcal{X}$:

$$\begin{aligned}
\langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x}_{t+1} - \boldsymbol{x}\rangle &\leq \langle \boldsymbol{\nabla}\omega(\boldsymbol{x}_{t+1}) - \boldsymbol{\nabla}\omega(\boldsymbol{x}_t), \boldsymbol{x} - \boldsymbol{x}_{t+1}\rangle \\
&= V_\omega(\boldsymbol{x}, \boldsymbol{x}_t) - V_\omega(\boldsymbol{x}, \boldsymbol{x}_{t+1}) - V_\omega(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t) && \text{Three-point identity.} \\
&\leq V_\omega(\boldsymbol{x}, \boldsymbol{x}_t) - V_\omega(\boldsymbol{x}, \boldsymbol{x}_{t+1}) - \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2. && \text{Fourth property of Bregman divergence.}
\end{aligned}$$

As a result,

$$\begin{aligned}
\gamma_t(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) &\leq \langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}^\star\rangle && \text{By definition of the subgradient.} \\
&= \langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\rangle + \langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle \\
&\leq V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_t) - V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_{t+1}) - \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\
&\quad + \langle \gamma_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle \\
&\leq V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_t) - V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_{t+1}) - \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\
&\quad + \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \frac{1}{2}\|\gamma_t \boldsymbol{g}_t\|_*^2 && \text{Young's inequality.} \\
&\leq V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_t) - V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_{t+1}) + \frac{\gamma_t^2}{2}\|\boldsymbol{g}_t\|_*^2.
\end{aligned}$$

■

Theorem 80 (Convergence of mirror descent)**.** Let f be convex and ω be 1-strongly convex on $\mathcal{X}$ w.r.t. norm $\|\cdot\|$. Then, mirror descent satisfies

$$\min_{t\in[T]} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq \frac{V_\omega(\boldsymbol{x}^\star, \boldsymbol{x}_0) + \frac{1}{2}\sum_{t=0}^{T-1}\gamma_t^2\|\boldsymbol{g}_t\|_*^2}{\sum_{t=0}^{T-1}\gamma_t}.$$

Note that this generalizes the convergence result of the subgradient method.

Suppose f is B-Lipschitz continuous such that $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq B\|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$. Namely, we then have $\|\boldsymbol{g}\|_* \leq B, \forall \boldsymbol{g} \in \partial f(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$.

Furthermore, let $R^2 \doteq \sup_{\boldsymbol{x}\in\mathcal{X}} V_\omega(\boldsymbol{x}, \boldsymbol{x}_0)$ and set

$$\gamma_t \doteq \frac{\sqrt{2}R}{B\sqrt{T}}.$$

Then, we have the following convergence rate,

$$\min_{t\in[T]} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq \mathcal{O}\left(\frac{BR}{\sqrt{T}}\right).$$

This is equivalent to the convergence rate of the subgradient method, but then for a more general notions of norm.

In practice, if we optimize over the simplex Δ^{d-1} with $\|\boldsymbol{g}\|_\infty \leq 1, \forall \boldsymbol{g} \in \partial f(\boldsymbol{x})$ and $\boldsymbol{x}_0 = [1/d, \ldots, 1/d]$. Then, we have the following convergence rate for the subgradient method, $\mathcal{O}\left(\frac{\sqrt{d}}{\sqrt{T}}\right)$, because $B \in \mathcal{O}\left(\sqrt{d}\right)$ and $R \in \mathcal{O}(1)$. On the other hand, we have the following convergence rate for mirror descent, $\mathcal{O}\left(\frac{\sqrt{\log d}}{\sqrt{T}}\right)$, since $B \in \mathcal{O}(1)$ and $R \in \mathcal{O}\left(\sqrt{\log d}\right)$. This is a considerable speedup.

12 Smoothing and proximal algorithms

Often, we want to optimize non-smooth functions. However, most of the time, we assume functions to be smooth. The question is thus whether we can exploit additional structure of non-smooth functions, instead of treating them as black boxes. The idea behind smoothing is to optimize a smooth approximation f_μ of the non-smooth function f.

12.1 Nesterov smoothing

Definition 81 (Conjugate function)**.** The conjugate function of f is

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathrm{dom}(f)} \Big\{\boldsymbol{x}^\top \boldsymbol{y} - f(\boldsymbol{x})\Big\}.$$

It is also called the Legendre-Fenchel transformation.

Lemma 82 (Convex conjugate properties)**.** The following holds for conjugate functions,

1. (Duality) If f is continuous and convex, then $f^{**} = f$;
2. (Fenchel's inequality)
$$f(\boldsymbol{x}) + f^*(\boldsymbol{y}) \geq \boldsymbol{x}^\top \boldsymbol{y}, \quad \forall \boldsymbol{x}, \boldsymbol{y};$$
3. If f and g are continuous and convex, then
$$(f+g)^*(\boldsymbol{x}) = \inf_{\boldsymbol{y}}\{f^*(\boldsymbol{y}) + g^*(\boldsymbol{x}-\boldsymbol{y})\};$$
4. If f is μ-strongly convex, then f^* is differentiable and $1/\mu$-smooth.

Nesterov smoothing approximates a non-smooth function f by

$$f_\mu(\boldsymbol{x}) = \max_{\boldsymbol{y}\in\mathrm{dom}(f^*)} \Big\{\boldsymbol{x}^\top \boldsymbol{y} - f^*(\boldsymbol{y}) - \mu \cdot d(\boldsymbol{y})\Big\},$$

where $d(\boldsymbol{y})$ is a proximity function. A proximity function is 1-strongly convex and non-negative. The function f_μ is $1/\mu$-smooth and approximates a convex f by

$$f(\boldsymbol{x}) - \mu D^2 \leq f_\mu(\boldsymbol{x}) \leq f(\boldsymbol{x}), \quad D^2 \doteq \max \boldsymbol{y} \in \mathrm{dom}(f^*) d(\boldsymbol{y}).$$

High μ results in a bad approximation.

Thus, we have a trade-off between approximation error and optimization efficiency. Specifically,

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) \leq \underbrace{f(\boldsymbol{x}) - f_\mu(\boldsymbol{x})}_{\textit{approximation error}} + \underbrace{f_\mu(\boldsymbol{x}) - \min_{\boldsymbol{x}} f_\mu(\boldsymbol{x})}_{\textit{optimization error}}.$$

The approximation error is on the order $\mathcal{O}(\mu)$, while the optimization error is on the order $\mathcal{O}(1/\mu t)$ using gradient descent.

If we apply accelerated gradient descent to solve the smoothed problem, we get an error of the following order,

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq \mathcal{O}\left(\mu D^2 + \frac{R^2}{\mu t^2}\right).$$

Note that this is faster than applying subgradient methods.

12.2 *Moreau-Yosida smoothing*

Moreau-Yosida regularization smooths f by

$$f_\mu(\boldsymbol{x}) = \min_{\boldsymbol{y} \in \mathrm{dom}(f^*)} \left\{ f(\boldsymbol{y}) + \frac{1}{2\mu}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \right\}.$$

This function is called the *Moreau envelope* of $f(\boldsymbol{x})$. For example, the Huber function is the Moreau envelope of $f(x) = |x|$,

$$f_\mu(x) = \begin{cases} \frac{x^2}{2\mu}, & |x| \leq \mu \\ |x| - \frac{\mu}{2}, & |x| > \mu. \end{cases}$$

As in Nesterov smoothing, f_μ is $1/\mu$-smooth. However, the advantage is that it minimizes exactly, *i.e.*, $\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \min_{\boldsymbol{x}} f_\mu(\boldsymbol{x})$.

12.3 *Proximal operators*

Definition 83 (Proximal operator)**.** The proximal operator of a convex function f at $\boldsymbol{x}$ is defined as

$$\mathrm{prox}_f(\boldsymbol{x}) \doteq \operatorname*{argmin}_{\boldsymbol{y} \in \mathrm{dom}(f)} \left\{ f(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \right\}.$$

Note that this is a special case of Moreau-Yosida regularization with $\mu = 1$. Furthermore, for many non-smooth functions, their proximal operator can be computed efficiently in a closed form.

12.4 *Proximal point algorithm*

The proximal point algorithm (PPA) repeatedly applies the proximal operator,

$$\boldsymbol{x}_{t+1} = \mathrm{prox}_{\lambda_t f}(\boldsymbol{x}_t).$$

Theorem 84 (Convergence of PPA)**.** If f is convex, then for any $T \geq 1$, we have

$$f(\boldsymbol{x}_{T+1}) - f(\boldsymbol{x}^\star) \leq \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{2\sum_{t=1}^T \lambda_t}.$$

If we set $\lambda_t = \lambda$ to be constant, we get a $\mathcal{O}(1/t)$ convergence rate.

12.5 Proximal gradient method

Assume the following convex composite optimization problem,

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

where f and g are convex.[13] The proximal gradient method (PGM) has the following update rule,

$$\boldsymbol{x}_{t+1} = \text{prox}_{\gamma_t g}(\boldsymbol{x}_t - \gamma_t \boldsymbol{\nabla} f(\boldsymbol{x}_t)).$$

Note that it alternates between a gradient update on f and a proximal operator on g.

Theorem 85 (Convergence of PGM)**.** Let $F(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x})$. Assume f is convex and L-smooth, g is convex and possibly non-smooth. Proximal gradient method with fixed stepsize $\gamma_t = 1/L$ satisfies

$$F(\boldsymbol{x}_t) - F(\boldsymbol{x}^\star) \leq \frac{L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{2t}.$$

This is nearly the same convergence rate as gradient descent, despite F being possibly non-smooth.

[13] Most supervised learning problems can be cast into this form,

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} \ell(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i) + g(\boldsymbol{\theta}),$$

where $h_{\boldsymbol{\theta}}$ is the predictor and g is a regularization function.

13 Stochastic optimization

Stochastic optimization involves decision-making in the presence of randomness. The optimization problem is formalized by a random vector $\boldsymbol{\xi} \sim P$,

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} F(\boldsymbol{x}) \doteq \mathbb{E}_{\boldsymbol{\xi}}[f(\boldsymbol{x}, \boldsymbol{\xi})].$$

P is unknown and can only be accessed through data, which make F and ∇F hard to compute. Furthermore, we assume unbiasedness of the stochastic gradient,

$$\mathbb{E}_{\boldsymbol{\xi}_t}[\nabla f(\boldsymbol{x}_t, \boldsymbol{\xi}_t) \mid \boldsymbol{x}_t] = \nabla F(\boldsymbol{x}_t).$$

In this setting, we use stochastic gradient descent (SGD), which has the following update rule,

$$\begin{aligned} \boldsymbol{\xi}_t &\overset{\text{iid}}{\sim} P \\ \boldsymbol{x}_{t+1} &= \boldsymbol{x}_t - \gamma_t \nabla f(\boldsymbol{x}_t, \boldsymbol{\xi}_t). \end{aligned}$$

13.1 Convergence analysis

In the non-convex case, we can show that SGD finds a stationary point with $\mathbb{E}\|\nabla F(\bar{\boldsymbol{x}})\| \leq \epsilon$ in $\mathcal{O}(1/\epsilon^4)$ gradient evaluations.

Theorem 86 (Non-convex, random output)**.** Suppose F is L-smooth and the stochastic gradient has bounded variance, *i.e.*, $\mathbb{E}\|\nabla f(\boldsymbol{x}, \boldsymbol{\xi}) - \nabla F(\boldsymbol{x})\|_2^2 \leq \sigma^2$. Then, SGD with $\gamma \doteq \min\{1/L, \gamma_0/\sigma\sqrt{T}\}$ achieves

$$\mathbb{E}\|\nabla F(\hat{\boldsymbol{x}}_T)\|^2 \leq \frac{\sigma}{\sqrt{T}}\left(\frac{2(F(\boldsymbol{x}_1) - F(\boldsymbol{x}^\star))}{\gamma_0} + L\gamma_0\right) + \frac{2L(F(\boldsymbol{x}_1) - F(\boldsymbol{x}^\star))}{T},$$

where $\hat{\boldsymbol{x}}_T \sim \text{Unif}(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\})$.

Proof.

$$\begin{aligned} \mathbb{E}[F(\boldsymbol{x}_{t+1}) - F(\boldsymbol{x}_t)] &\leq \mathbb{E}\left[\nabla F(\boldsymbol{x}_t)^\top(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t) + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|^2\right] && \text{Smoothness of } F. \\ &= \mathbb{E}\left[-\gamma_t \nabla F(\boldsymbol{x}_t)^\top \nabla f(\boldsymbol{x}_t, \boldsymbol{\xi}_t) + \frac{L\gamma_t^2}{2}\|\nabla f(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\|^2\right] && \text{SGD update rule.} \\ &= -(\gamma_t - \frac{L\gamma_t^2}{2})\mathbb{E}\|\nabla F(\boldsymbol{x}_t)\|^2 + \frac{L\sigma^2\gamma_t^2}{2} && \mathbb{E}[X^2] = \mathbb{E}[X]^2 + \mathbb{V}[X]. \\ &\leq -\frac{\gamma_t}{2}\mathbb{E}\|\nabla F(\boldsymbol{x}_t)\|^2 + \frac{L\sigma^2\gamma_t^2}{2}. && \gamma_t \leq 1/L. \end{aligned}$$

We can rewrite this as

$$\mathbb{E}\|\nabla F(\boldsymbol{x}_t)\|^2 \leq \frac{2\mathbb{E}[F(\boldsymbol{x}_t) - F(\boldsymbol{x}_{t+1})]}{\gamma_t} + \gamma_t\sigma^2 L.$$

By definition of $\hat{\boldsymbol{x}}_T$, we have

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{\nabla} F(\hat{\boldsymbol{x}}_T)\|^2 &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{\nabla} F(\boldsymbol{x}_t)\|^2 \\
&\leq \frac{1}{T}\left(\sum_{t=1}^{T}\frac{2\mathbb{E}[F(\boldsymbol{x}_t)-F(\boldsymbol{x}_{t+1})]}{\gamma_t}+\gamma_t\sigma^2 L\right) \\
&= \frac{2}{\gamma T}\left(\sum_{t=1}^{T}F(\boldsymbol{x}_t)-F(\boldsymbol{x}_{t+1})\right)+\gamma\sigma^2 L && \text{Constant stepsize.} \\
&= \frac{2(F(\boldsymbol{x}_1)-F(\boldsymbol{x}_{T+1}))}{\gamma T}+\gamma\sigma^2 L && \text{Telescoping sum.} \\
&\leq \frac{2(F(\boldsymbol{x}_1)-F(\boldsymbol{x}^\star))}{\gamma T}+\gamma\sigma^2 L \\
&\leq \frac{2(F(\boldsymbol{x}_1))-F(\boldsymbol{x}^\star)}{T}\max\left\{L,\frac{\sigma\sqrt{T}}{\gamma_0}\right\}+\frac{\gamma_0\sigma L}{\sqrt{T}} \\
&\leq \frac{2L(F(\boldsymbol{x}_1)-F(\boldsymbol{x}^\star))}{T}+\frac{\sigma}{\sqrt{T}}\left(\frac{2(F(\boldsymbol{x}_1)-F(\boldsymbol{x}^\star))}{\gamma_0}+L\gamma_0\right).
\end{aligned}$$

■

In the convex case, we can show that SGD finds an ϵ-optimal solution with $\mathcal{O}(1/\epsilon^2)$ sample complexity.

Theorem 87 (Convex, weighted averaging)**.** Suppose F is convex and $\mathbb{E}\|\boldsymbol{\nabla} f(\boldsymbol{x},\boldsymbol{\xi})\|^2 \leq B^2, \forall \boldsymbol{x}$. Then, SGD satisfies

$$\mathbb{E}[F(\hat{\boldsymbol{x}}_T)-F(\boldsymbol{x}^\star)] \leq \frac{R^2+B^2\sum_{t=0}^{T}\gamma_t^2}{2\sum_{t=0}^{T}\gamma_t},$$

where

$$\hat{\boldsymbol{x}}_T \doteq \frac{\sum_{t=0}^{T}\gamma_t\boldsymbol{x}_t}{\sum_{t=0}^{T}\gamma_t}, \quad \|\boldsymbol{x}_0-\boldsymbol{x}^\star\| \leq R.$$

Proof. First, we have

$$\begin{aligned}
\|\boldsymbol{x}_{t+1}-\boldsymbol{x}^\star\|^2 &= \|\boldsymbol{x}_t-\gamma_t\boldsymbol{\nabla} f(\boldsymbol{x}_t,\boldsymbol{\xi}_t)-\boldsymbol{x}^\star\|^2 \\
&= \|\boldsymbol{x}_t-\boldsymbol{x}^\star\|+\gamma_t^2\|\boldsymbol{\nabla} f(\boldsymbol{x}_t,\boldsymbol{\xi}_t)\|^2-2\boldsymbol{\nabla} f(\boldsymbol{x}_t,\boldsymbol{\xi}_t)^\top(\boldsymbol{x}_t-\boldsymbol{x}^\star). && \text{Cosine theorem.}
\end{aligned}$$

Furthermore, by the law of total expectation ($\mathbb{E}[X]=\mathbb{E}_Y[\mathbb{E}_X[X\mid Y]]$),

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}_{1:t}}\left[\boldsymbol{\nabla} f(\boldsymbol{x}_t,\boldsymbol{\xi}_t)^\top(\boldsymbol{x}_t-\boldsymbol{x}^\star)\right] &= \mathbb{E}_{\boldsymbol{\xi}_{1:t-1}}\left[\mathbb{E}_{\boldsymbol{\xi}_t}\left[\boldsymbol{\nabla} f(\boldsymbol{x}_t,\boldsymbol{\xi}_t)^\top(\boldsymbol{x}_t-\boldsymbol{x}^\star)\;\middle|\;\boldsymbol{x}_t\right]\right] \\
&= \mathbb{E}_{\boldsymbol{\xi}_{1:t-1}}\left[\mathbb{E}_{\boldsymbol{\xi}_t}[\boldsymbol{\nabla} f(\boldsymbol{x}_t,\boldsymbol{\xi}_t)\mid\boldsymbol{x}_t]^\top(\boldsymbol{x}_t-\boldsymbol{x}^\star)\right] && \boldsymbol{x}_t \text{ can be computed from } \boldsymbol{\xi}_{1:t-1}. \\
&= \mathbb{E}_{\boldsymbol{\xi}_{1:t-1}}\left[\boldsymbol{\nabla} F(\boldsymbol{x}_t)^\top(\boldsymbol{x}_t-\boldsymbol{x}^\star)\right] \\
&\geq \mathbb{E}[F(\boldsymbol{x}_t)-F(\boldsymbol{x}^\star)]. && \text{Convexity of } F.
\end{aligned}$$

This gives us the following recursion,

$$\gamma_t\mathbb{E}[F(\boldsymbol{x}_t)-F(\boldsymbol{x}^\star)] \leq \frac{1}{2}\mathbb{E}\|\boldsymbol{x}_t-\boldsymbol{x}^\star\|^2-\frac{1}{2}\mathbb{E}\|\boldsymbol{x}_{t+1}-\boldsymbol{x}^\star\|^2+\frac{1}{2}\gamma_t^2B^2,$$

and the result follows by telescoping the sum from $t = 1$ to T. ■

In the strongly convex case, we can show that SGD finds an ϵ-optimal solution with $\mathcal{O}(1/\epsilon)$ complexity.

Theorem 88 (Strongly convex, diminishing stepsize)**.** Suppose F is μ-strongly convex and $\mathbb{E}\|\nabla f(\boldsymbol{x}, \boldsymbol{\xi})\|^2 \leq B^2, \forall \boldsymbol{x}$, then SGD with $\gamma_t = \frac{\gamma}{t}$ and $\gamma > 1/2\mu$ satisfies

$$\mathbb{E}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 \leq \frac{C(\gamma)}{t},$$

where

$$C(\gamma) = \max\left\{\frac{\gamma^2 B^2}{2\mu\gamma - 1}, \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2\right\}.$$

Proof. Like in the proof of the previous case, we have

$$\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 = \|\boldsymbol{x}_t - \boldsymbol{x}^\star\| + \gamma_t^2\|\nabla f(\boldsymbol{x}_t, \boldsymbol{\xi}_t)\|^2 - 2\nabla f(\boldsymbol{x}_t, \boldsymbol{\xi}_t)^\top(\boldsymbol{x}_t - \boldsymbol{x}^\star).$$

Also like in the previous proof and further using strong convexity of F, we have

$$\mathbb{E}\Big[\nabla f(\boldsymbol{x}_t, \boldsymbol{\xi}_t)^\top(\boldsymbol{x}_t - \boldsymbol{x}^\star)\Big] = \mathbb{E}\Big[\nabla F(\boldsymbol{x}_t)^\top(\boldsymbol{x}_t - \boldsymbol{x}^\star)\Big] \geq \mu\mathbb{E}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2.$$

This gives the following recursion,

$$\mathbb{E}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \leq \left(1 - \frac{2\mu\gamma}{t}\right)\mathbb{E}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 + \frac{\gamma^2 B^2}{t^2}.$$

The results follows by induction. ■

Thus, in theory, we see that a diminishing stepsize is necessary for SGD to converge to an optimal solution. However, in practice, constant stepsizes are often used with great success.

13.2 Adaptive methods

Often we do not know whether the problem is convex, L-smooth, or μ-strongly convex. Thus, we want the stepsize to adapt to the landscape of the function. The generic adaptive scheme looks like the following,

$$\begin{aligned}
\boldsymbol{g}_t &= \nabla f(\boldsymbol{x}_t, \boldsymbol{\xi}_t) \\
\boldsymbol{m}_t &= \phi_t(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_t) && \text{Momentum.} \\
\boldsymbol{V}_t &= \psi_t(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_t) \\
\hat{\boldsymbol{x}}_t &= \boldsymbol{x}_t - \alpha_t \boldsymbol{V}_t^{-1/2}\boldsymbol{m}_t \\
\boldsymbol{x}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{x} \in X}\Big\{(\boldsymbol{x} - \hat{\boldsymbol{x}}_t)^\top \boldsymbol{V}_t^{1/2}(\boldsymbol{x} - \hat{\boldsymbol{x}}_t)\Big\}.
\end{aligned}$$

The most popular stochastic gradient descent methods are special cases of this scheme,

- Stochastic gradient descent:

$$\boldsymbol{m}_t = \boldsymbol{g}_t, \quad \boldsymbol{V}_t = \boldsymbol{I}.$$

- AdaGrad:

$$\boldsymbol{m}_t = \boldsymbol{g}_t, \quad \boldsymbol{V}_t = \frac{\operatorname{diag}\left(\sum_{\tau=1}^{t} \boldsymbol{g}_\tau^2\right)}{t}.$$

- Adam:

$$\boldsymbol{m}_t = (1-\alpha)\sum_{\tau=1}^{t} \alpha^{t-\tau}\boldsymbol{g}_\tau, \quad \boldsymbol{V}_t = (1-\beta)\operatorname{diag}\left(\sum_{\tau=1}^{t} \beta^{t-\tau}\boldsymbol{g}_\tau^2\right).$$

  Or, recursively:

$$\boldsymbol{m}_t = \alpha\boldsymbol{m}_{t-1} + (1-\alpha)\boldsymbol{g}_t, \quad \boldsymbol{V}_t = \beta\boldsymbol{V}_{t-1} + (1-\beta)\operatorname{diag}\left(\boldsymbol{g}_t^2\right).$$

13.3 Variance reduction

Despite having a cheaper iteration cost than gradient descent,[14] SGD requires more iterations,[15] due to high variance. Stochastic variance-reduced (VR) methods try to achieve the best of both worlds by reducing the variance of SGD.[16] We will present VR methods in the context of finite-sum optimization, which is a special case of stochastic optimization,

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}) \doteq \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}).$$

In the context of deep learning, we can see n as the number of data points and f_i the function w.r.t. the i-th data point, where we wish to minimize the objective function w.r.t. every data point with equal weight.

Suppose we want to estimate $\theta = \mathbb{E}[X]$, where X is a random variable. Let Y be another random variable. We can estimate θ as $\mathbb{E}[X-Y]$ if and only if $\mathbb{E}[Y] = 0$. Furthermore, $\mathbb{V}[X-Y] \leq \mathbb{V}[X]$ if Y is highly positively correlated with X. Specifically, if $\operatorname{Cov}(X,Y) > \frac{1}{2}\mathbb{V}[Y]$, the variance will be reduced.[17]

Let $\alpha \in [0,1]$. Using the following point estimator introduces a trade-off between variance and biasedness,

$$\hat{\theta}_\alpha = \alpha(X-Y) + \mathbb{E}[Y].$$

We then have the following expected value and variance,

$$\begin{aligned}\mathbb{E}[\hat{\theta}_\alpha] &= \alpha\mathbb{E}[X] + (1-\alpha)\mathbb{E}[Y]\\ \mathbb{V}[\hat{\theta}_\alpha] &= \alpha^2(\mathbb{V}[X] + \mathbb{V}[Y] - 2\operatorname{Cov}(X,Y)).\end{aligned}$$

Note that the estimator is unbiased if $\alpha = 1$, but the variance decreases when α decreases.

[14] $\mathcal{O}(1)$ for SGD *vs.* $\mathcal{O}(n)$ for GD.

[15] $\mathcal{O}(\kappa/\epsilon)$ for SGD *vs.* $\mathcal{O}(\kappa \log 1/\epsilon)$ for GD, where $\kappa = L/\mu$.

[16] Classically, one can reduce variance by mini-batching, which reduces variance by $\mathcal{O}(1/|B_t|)$, where B_t is the batch, but computational complexity increases by $\mathcal{O}(|B_t|)$. Variance can also be reduced by introducing momentum to the gradient step. However, this requires access to past stochastic gradients, which can be expensive in memory.
We will consider a more modern approach.

[17] This is because $\mathbb{V}[X-Y] = \mathbb{V}[X] + \mathbb{V}[Y] - 2\operatorname{Cov}(X,Y)$.

While SGD estimates the full gradient by $\nabla f_{i_t}(x_t)$, VR methods estimate $\nabla F(x_t)$ by

$$g_t \doteq \alpha(\nabla f_{i_t}(x_t) - Y) + \mathbb{E}[Y],$$

such that

$$\mathbb{E}\|g_t - \nabla F(x_t)\|^2 \overset{t\to\infty}{\to} 0.$$

VR property.

The key idea is that if x_t is not too far away from previous iterates $x_{1:t-1}$, then we can leverage previous gradient information to construct positively correlated control variates Y. The question is thus how to design Y, given previous gradient information, such that it has low computational and space complexity.

Stochastic average gradient. The idea behind stochastic average gradient (SAG) is to keep track of the latest gradients for all points $i \in [n]$. Then, we estimate the full gradient by the average of these recent gradients,

$$g_t = \frac{1}{n}\sum_{i=1}^{n} v_i^t \quad \approx \quad \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_t) = \nabla F(x_t).$$

Thus, we update the past gradients as

$$v_i^t = \begin{cases} \nabla f_{i_t}(x_t) & i = i_t \\ v_i^{t-1} & i \neq i_t. \end{cases}$$

Equivalently, we thus have the following update rule,

$$\begin{aligned} g_t &= g_{t-1} - \frac{1}{n} v_{i_t}^{t-1} + \frac{1}{n}\nabla f_{i_t}(x_t) \\ &= \frac{1}{n}\left(\nabla f_{i_t}(x_t) - v_{i_t}^{t-1}\right) + g_{t-1}. \end{aligned}$$

Thus, we have $\alpha = 1/n$ and $Y = v_{i_t}^{t-1}$ with $\mathbb{E}[Y] = g_{t-1}$.

The downside of this approach is that it has a biased gradient, a large $\mathcal{O}(nd)$ memory cost, and it is hard to analyze. But, we gain a total complexity of $\mathcal{O}((n + \kappa_{\max}) \log 1/\epsilon)$, where $\kappa_{\max} = \min_{i\in[n]} L_i/\mu$, where L_i is the smoothness parameter of f_i.

SAGA is an unbiased version of SAG, because it sets $\alpha = 1$, but still enjoys the same benefits as SAG with a much simpler proof. However, we still have a higher memory cost than SGD, which we would like to get rid of.

Stochastic variance reduced gradient. The key idea behind stochastic variance reduced gradient (SVRG) is to build covariates based on a fixed reference point $\tilde{x}$. We then need to balance the frequency of updating this reference point and variance reduction.[18] The intuition behind this method is the closer $\tilde{x}$ is to x_t, the smaller the variance is of the gradient estimator.

[18] More updates cause lower variance, but increased complexity.

Algorithm	Iterations	Iteration cost
Gradient descent	$\mathcal{O}(\kappa \log 1/\epsilon)$	$\mathcal{O}(n)$
Stochastic gradient descent	$\mathcal{O}(\kappa_{\max}/\epsilon)$	$\mathcal{O}(1)$
Variance-reduced method	$\mathcal{O}((n+\kappa_{\max}) \log 1/\epsilon)$	$\mathcal{O}(1)$

Table 1. Complexity of μ-strongly convex and L-smooth finite-sum optimization, where n is the number of functions, $\kappa = L/\mu$, and $\kappa_{\max} = \max_{i\in[n]} L_i/\mu$, where L_i is the smoothness parameter of f_i.

The algorithm works by updating $\tilde{\boldsymbol{x}}$ every m-th update and estimates the gradient by

$$\boldsymbol{\nabla} f_{i_t}(\boldsymbol{x}_t) - \boldsymbol{\nabla} f_{i_t}(\tilde{\boldsymbol{x}}) + \boldsymbol{\nabla} F(\tilde{\boldsymbol{x}}).$$

Thus, we have $\alpha = 1$ and $Y = \boldsymbol{\nabla} f_{i_t}(\tilde{\boldsymbol{x}})$ with $\mathbb{E}[Y] = \boldsymbol{\nabla} F(\tilde{\boldsymbol{x}})$.

While we gain the low memory cost of $\mathcal{O}(d)$, we now need to do $\mathcal{O}(n + 2m)$ gradient evaluations per epoch, where the n comes from computing $\mathbb{E}[Y]$ and $2m$ comes from computing $\boldsymbol{\nabla} f_{i_t}(\boldsymbol{x}_t)$ and $\boldsymbol{\nabla} f_{i_t}(\tilde{\boldsymbol{x}})$. This method has the same complexity as SAG and SAGA.

References

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.

John Greenstadt. Variations on variable-metric methods. *Mathematics of Computation*, 24(109):1–22, 1970.

Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Julie Nutini, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641. PMLR, 2015.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1): 267–288, 1996.