Gradient descent and quasi-Newton methods

Lecturer: Žiga Virk

Course: Mathematics 2

March 25, 2024

The main goal is to present the gradient descent (GD) procedure, corresponding theoretical guarantees and a series of modifications of GD (mostly based on [1] and [2]). We conclude by Newton and quasi-Newton methods (mostly based on [3]).

Contents

1 Introduction **2**

2 Properties of functions **4**

Convex functions 4

Lipschitz functions 5

Smooth functions 6

Strongly convex functions 7

Summary 8

3 Guarantees for gradient descent **9**

Proofs 11

Summary 13

4 Lower bounds for black box procedures **13**

5 Modifications of gradient descent **16**

Quadratics 16

Polyak (also called momentum or heavy ball) GD 17

Nesterov GD 19

Stochastic GD 20

AdaGrad GD 21

Summary 23

6 Newton and quasi-Newton methods **23**

Newton methods 24

Adaptations to the Newton method 26

Quasi-Newton methods 27

Limited memory quasi-Newton methods 29

Summary 31

1 Introduction

MAIN TASK: given a function f find its minimum[1] x^*, i.e., find x^* from the domain of f minimizing $f(x)$.

MAIN STRATEGY: Iterative method, i.e., start with some initial guess x_1 and inductively keep making educated guesses about the next step, hoping that x_k converge to x^*.

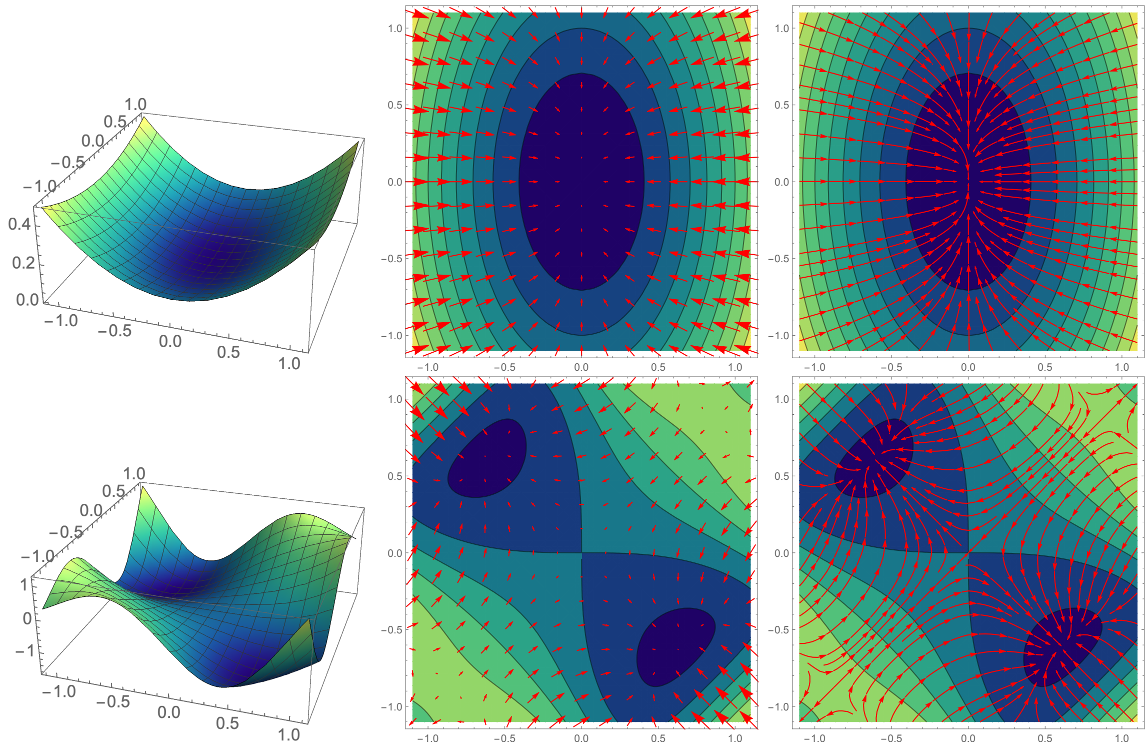

Figure 1: The first row shows a graph of function $f(x, y) = 0.3x^2 + 0.1y^2$, its contour plot (level lines) with the $-\nabla f$ gradient field, and its contour plot with stream lines (lines following the gradient). Note that the stream lines always end at the minimum. This is the effect we want to replicate in a discrete setting with GD. Bottom row shows the same effect for $\sin(3xy) - \cos\left(x^2 + y^2\right)$. Note that this function has two local minima in the interior of the displayed region and six local minima on the boundary. A stream line can end up in any stationary point.

Depending on the nature of information used for guessing we have:

- zero-order methods: we use f (bisection, secant method, ...).
- first-order methods: we use f and ∇f (GD, Quasi-Newton, ...).
- second-order methods: we use $f, \nabla f$, and Hessian $\nabla^2 f = Hf$ (Newton method, ...).

We will first focus on GD. Let $f\colon D \to \mathbb{R}$ be a differentiable function with $D \subset \mathbb{R}^n$. The idea is that ∇f points at the direction of the greatest ascent of f, hence going along the direction $-\nabla f$ the function is decreasing at the highest rate compared to all other directions. We first choose an initial guess $x_1 \in D$ and set a parameter $\gamma > 0$ (sometimes called learning rate). We then proceed as follows:

GRADIENT descent GD:
$$x_{k+1} = x_k - \gamma \nabla f(x_k)$$

The differentiability assumtion could be relaxed a bit for convex functions using subgradients (see Bubeck's book). In this case GD as described below still mostly works.

[1] Typically we can't find it analytically or an analytic solution is too time consuming to compute. We will usually consider convex functions with a unique minimum.

Example 1.1. *Let $f(x) = x^2$ (with the obvious minimum at $x = 0$) and choose $x_0 = 1$.*

- *If $\gamma = 1$ then we get a sequence of approximations $1, -1, 1, -1, \ldots$ so GD does not converge but enters a cyclic behaviour.*
- *For $\gamma > 1$ it is easy to see that $|x_i| \to \infty$.*
- *For $1/2 < \gamma < 1$ GD converges (i.e., $x_i \to 0$) in an alternating way, i.e., in each step the sign of a new approximation x_i is changed.*
- *For $\gamma = 1/2$ GD converges in one step, i.e., $x_2 = 0$.*
- *For $\gamma < 1/2$ GD converges in a monotonous way, i.e., all x_i are positive.*

From Example 1.1 and Figure 1 it is clear that GD does not always converge[2], that the convergence depends on x_1 and γ, and that theory and practice are needed for its successful application. Here is a fundamental question we will try to partially answer: In what settings does GD converge and how fast does it converge? To begin with, we will need to restrict to functions with additional properties.

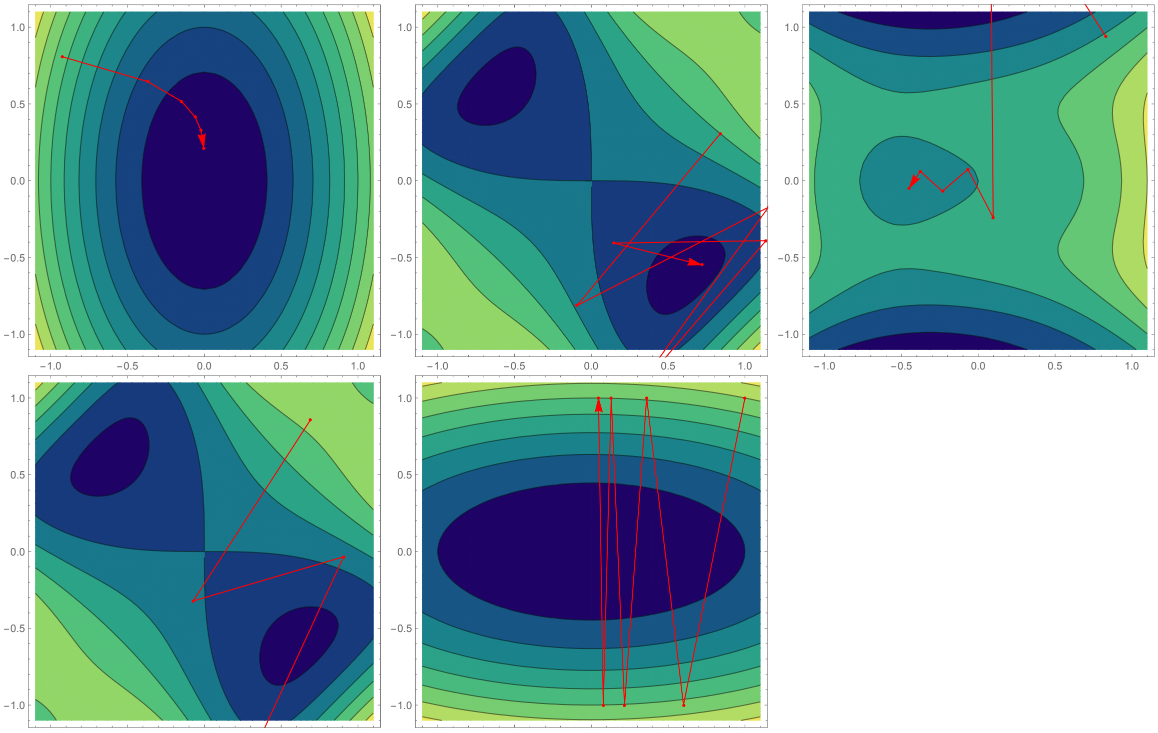

Figure 2: A few examples of convergence (top) and divergence (bottom) of GD.

[2] Even when GD converges the limiting value may not be a local minimum. For example, consider an analogy of case $\gamma < 1/3$ of Example 1.1 for function x^3: we will get $x_i \to 0$, which is not a local minimum.

2 Properties of functions

Throughout these notes $D \subset R^n$ and $f\colon D \to \mathbb{R}$ is a continuous differentiable function. In this section we introduce properties of functions that will allow us to deduce convergence guarantees.

Convex functions

> **Definition 2.1.** *A subset D is* ***convex*** *if $\forall x, y \in D, \forall t \in [0,1]$ we have $tx + (1-t)y \in D$, i.e., if $\forall x, y \in D$ the whole line segment x to y is in D.*

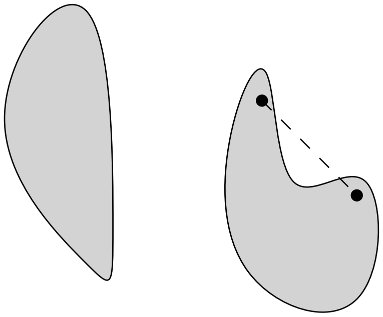

Figure 3: A convex(left) and a non-convex set (right).

> **Definition 2.2.** *Function $f\colon D \to \mathbb{R}$ is* ***convex*** *if D is convex and $\forall x, y \in D, \forall t \in [0,1]$ we have*
>
> $$f(tx + (1-t)y) \le tf(x) + (1-t)f(y).$$

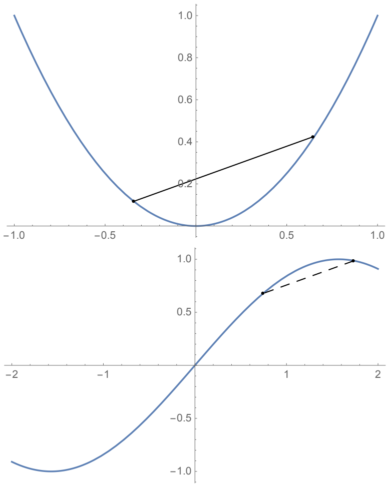

Figure 4: A convex(top) and a non-convex function (bottom).

Convex functions include $x^2, x^{2020}, e^x, -\log(x), -\sqrt{x}$. Non-convex functions include $\sin(x), \cos(x), x^3, \log(x)$.

> **Proposition 2.3.** *Let f be a function and let Γ be its graph. The following are equivalent:*
>
> 1. *f is convex.*
> 2. *For each $x, y \in D$ the line segment from $(x, f(x))$ to $(y, f(y))$ lies above Γ.*
> 3. *For each $x \in D$ the tangent hyperplane to Γ at x lies below Γ, i.e., $\forall x, y \in D$:*
> $$f(y) \ge f(x) + \nabla f(x)^T (y - x).$$
> 4. *For each $x_1, x_2, \ldots, x_k \in D$ and $\alpha_1, \alpha_2, \ldots, \alpha_k \in [0,1]$ satisfying $\sum_{i=1}^k \alpha_i = 1$ we have*
> $$f\Big(\sum_{i=1}^k \alpha_i x_i\Big) \le \sum_{i=1}^k \alpha_i f(x_i).$$
>
> *Furthermore, if f is twice continuously differentiable then f is convex iff all eigenvalues of $\nabla^2 f$ are non-negative (equivalently, if $\nabla^2 f$ is positively semidefinite).*

Given $x, y \in \mathbb{R}^n$ their dot product will be denoted by $x^T y = \langle x, y \rangle$. The norm of x will be denoted by $||x||$.

Relations "above" and "below" in Proposition 2.3 hold in a non-strict sense, i.e., below graph Γ means strictly below or on graph Γ, as the non-strict inequalities suggest. A version of Proposition 2.3 in a strict sense holds for strictly convex functions defined below.

Proof. Equivalence 1. ⇔ 2. holds by definition. Equivalence 1. ⇔ 4. can be proved using induction. For the other two equivalences see https://wiki.math.ntnu.no/_media/tma4180/2016v/note2.pdf. □

> **Proposition 2.4.** *If f is a convex function the following hold:*
>
> 1. *For each $h \in \mathbb{R}$ the sublevel set $f^{-1}((-\infty, h])$ is convex.*
> 2. *Each local minimum is a global minimum.*
> 3. *The set of global minima of f is a convex set.*

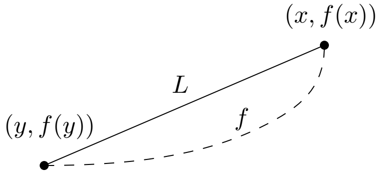

Figure 5: A sketch of the proof of 2. of Proposition 2.4.

Proof. 1. follows directly from definition. In order to prove 2. let $x, y \in D$ be two local minima of f with $f(x) > f(y)$. Since the graph of f lies below the line segment L from $(x, f(x))$ to $(y, f(y))$ we conclude that x can't be a local minimum, see Figure 5. In order to prove 3. let $x, y \in D$ be two global minima of f. The same argument as for 2. shows that since $f(x) = f(y)$, the function value along the line segment L from $(x, f(x))$ to $(y, f(y))$ is $f(x)$, hence all points of L are global minima. □

The converse of 1. of Proposition 2.4 does not hold. Think about function $\sqrt{|x|}$

Most of our functions will actually be **strictly convex**, that is $\forall x, y \in D, \forall t \in (0, 1)$ the following holds:

$$f(tx + (1-t)y) < tf(x) + (1-t)f(y).$$

In such a case it is easy to prove f has at most one local[3] minimum[4].

Lipschitz functions

> **Definition 2.5.** *Let $L > 0$. Function f is L-**Lipschitz** if $\forall x, y \in D$ we have*
> $$|f(x) - f(y)| \leq L||x - y||.$$

1-Lipschitz functions include $\sin(x), \cos(x), \arctan(x), |x|$. Functions, which are not L-Lipschitz on their entire domain but are L-Lipschitz (for some L) on each bounded interval include x^n for $n > 1$ and e^x. Logarithmic function fails to be L-Lipschitz even on $(0, 1)$ and $\sqrt{x}$ fails to be L-Lipschitz even on $[0, 1]$.

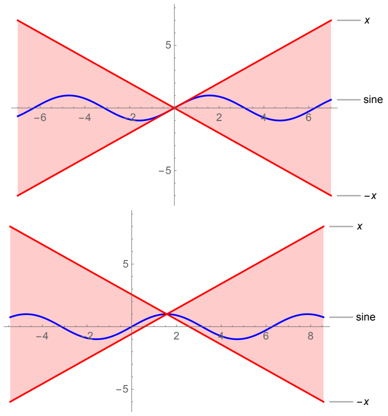

Figure 6: Function $f(x) = \sin(x)$ is 1-Lipschitz because for each point p on the graph Γ of f, Γ is between the linear functions with slopes ±1 passing through p .

> **Proposition 2.6.** *Let $L > 0$.*
>
> 1. *If f is L-Lipschitz, it is continuous.*
> 2. *f is L-Lipschitz iff $||\nabla f|| \leq L$.*

Proof. 1.
$$\lim_{x_n \to x} |f(x_n) - f(x)| \leq L \lim_{x_n \to x} ||x_n - x|| = 0.$$

[3] And hence global by Proposition 2.4.

[4] This set might be empty, for example in the case of e^x.

2. Assume f is L-Lipschitz. For[5] $h = \frac{\nabla f(x)}{||\nabla f(x)||}$ we can use directional derivative:

$$||\nabla f(x)|| = |h^T \nabla f(x)| = \lim_{t \searrow 0} \frac{|f(x+th) - f(x)|}{t} \leq \lim_{t \searrow 0} \frac{L||th||}{t} = L.$$

Conversely if $||\nabla f|| \leq L$ we define $h = \frac{y-x}{||y-x||}$ and deduce

$$|f(x) - f(y)| = |\int_0^{||x-y||} f'(x+th)dt| \leq \int_0^{||x-y||} |f'(x+th)|dt \leq$$

$$\leq \int_0^{||x-y||} Ldt = L||x-y||,$$

where the derivatives f' are with respect to variable t. □

Smooth functions

> **Definition 2.7.** *Let $\beta > 0$. Function f is **β-smooth** if $\forall x, y \in D$:*
>
> $$||\nabla f(x) - \nabla f(y)|| \leq \beta ||x - y||.$$

Essentially, function f is β-smooth iff ∇f is β-Lipschitz.

β-smooth functions include $\sin(x), \cos(x), \arctan(x), |x|$ and quadratic functions. Functions, which are not β-smooth on their entire domain but are β-smooth (for some β) on each bounded interval include x^n for $n > 1$ and e^x. Logarithmic function fails to be β-smooth even on $(0, 1)$ and $\sqrt{x}$ fails to be β-smooth even on $[0, 1]$.

Function x^2 is β-smooth but not L-Lipschitz. Function $x^{3/2}$ restricted to $[0, 1]$ is L-Lipschitz but not β-smooth.

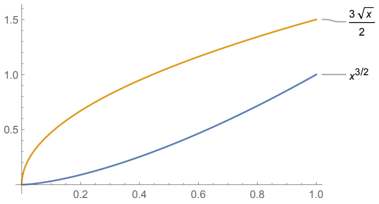

Figure 7: Function $f(x) = x^{3/2}$ and its derivative. It is clear that f is Lipschitz but not β-smooth as f' is not Lipschitz.

> **Proposition 2.8.** *Let $\beta > 0$. Suppose f is a twice continuously differentiable convex function. Then the following are equivalent:*
>
> *1. f is β-smooth.*
>
> *2. $||\nabla^2 f|| \leq \beta$.*
>
> *3. All eigenvalues of $\nabla^2 f$ lie on $[0, \beta]$.*

Proof. Equivalence 1. ⇔ 2. is a multivariate version[6] of statement 2. of Proposition 2.6.

Equivalence 2. ⇔ 3.: Because $\nabla^2 f$ is symmetric[7] its eigenvalues are also its singular values up to the sign. $\nabla^2 f$ is also positive semidefinite[8], meaning all its eigenvalues are non-negative. Since $||\nabla^2 f||$ is the largest singular value the equivalence holds. □

Note that $\nabla^2 f$ depends on x and so do its eigenvalues and $||\nabla^2 f||$, meaning that conditions 2. and 3. of Proposition 2.8 take into account all $x \in D$.

[5] If $||\nabla f(x)|| = 0$ the statement holds trivially.

[6] If $n = 1$ is is actually the same version.

[7] Hessian is always symmetric.

[8] As f is convex.

Proposition 2.9 (Lemmas 3.4 and 3.5 in Bubeck)**.** *Let $\beta > 0$. Suppose f is convex and β-smooth. Then for any $x, y \in D$:*

1.

$$|f(x) - f(y) - \nabla f(y)^T(x-y)| \le \frac{\beta}{2}||x-y||^2$$

2.

$$f(x) - f(y) \le \nabla f(x)^T(x-y) - \frac{1}{2\beta}||\nabla f(x) - \nabla f(y)||^2.$$

Proof. 1.

$$\begin{aligned}
&|f(x) - f(y) - \nabla f(y)^T(x-y)| &&= \\
&= \Big| \int_0^1 \nabla f(y + t(x-y))^T(x-y)dt - \nabla f(y)^T(x-y) \Big| &&= \\
&= \Big| \int_0^1 (\nabla f(y + t(x-y))^T - \nabla f(y)^T)(x-y)dt \Big| &&\le \\
&\le \int_0^1 ||(\nabla f(y + t(x-y))^T - \nabla f(y)^T)|| \cdot ||x-y||dt &&\le \\
&\le \int_0^1 \beta t||x-y|| \cdot ||x-y||dt = \frac{\beta}{2}||x-y||^2.
\end{aligned}$$

$$f(x) - f(y) =$$
$$= \int_0^1 \nabla f(y + t(x-y))^T(x-y)dt$$

2. Define $z = y - \frac{1}{\beta}(\nabla f(y) - \nabla f(x))$.

$$\begin{aligned}
&f(x) - f(y) = (f(x) - f(z)) + (f(z) - f(y)) \le \\
&= \nabla f(x)^T(x-z) + \nabla f(y)^T(z-y) + \frac{\beta}{2}||z-y||^2 = \\
&= \nabla f(x)^T(x-y) + (\nabla f(x) - \nabla f(y))^T(y-z) + \frac{1}{2\beta}||\nabla f(x) - \nabla f(y)||^2 = \\
&= \nabla f(x)^T(x-y) - \frac{1}{2\beta}||\nabla f(x) - \nabla f(y)||^2.
\end{aligned}$$

□

Let us provide a few hints for the proof of (2) of Proposition 2.9. By (3) of Proposition 2.3:

$$f(x) - f(z) \le \nabla f(x)^T(x-z).$$

By (1) of Proposition 2.9:

$$f(z) - f(y) \le \nabla f(y)^T(z-y) + \frac{\beta}{2}||z-y||^2.$$

In the last two lines insert the expression of z for the right instance first, and then for the left instance.

Strongly convex functions

Definition 2.10. *Let $\alpha > 0$. Function f is α-**strongly convex** if $f(x) - \frac{\alpha}{2}||x||^2$ is convex. Informally speaking, f is α-strongly convex if it is more convex than $\frac{\alpha}{2}||x||^2$.*

Proposition 2.11. *Let $\alpha > 0$.*

1. *Function f is α-strongly convex iff $\forall x, y \in D$:*

$$f(x) - f(y) \le \nabla f(x)^T(x-y) - \frac{\alpha}{2}||x-y||^2.$$

> 2. *A twice continuously differentiable function f is α-strongly convex iff each eigenvalue of $\nabla^2 f$ is greater or equal to α.*

Proof. 1. Using 3.[9] of Proposition 2.3 we see that f is α-strongly convex iff

$$f(y) - \frac{\alpha}{2}||y||^2 \geq f(x) - \frac{\alpha}{2}||x||^2 + (\nabla f(x) - \alpha x)^T(y-x),$$

which can be rearranged into

$$-\frac{\alpha}{2}(||x||^2 + ||y||^2 - 2x^Ty) + \nabla f(x)^T(x-y) \geq f(x) - f(y)$$

and noting $||x||^2 + ||y||^2 - 2x^Ty = ||x-y||^2$ we are done.

2. Follows from the last part[10] of Proposition 2.3 using the equality $\nabla^2(f(x) - \frac{\alpha}{2}||x||^2) = \nabla^2 f(x) - \alpha I$. □

Summary

We presented four properties of functions that will allow us to prove theoretical guarantees[11]:

- Convex functions[12] represent a standard class of functions for which a nice optimization theory can be developed. They are typically strongly convex, meaning they have at most one minimum. They are bounded below by their tangents:

$$f(x) \geq f(z) + \nabla f(z)^T(x-z).$$

- L-Lipschitz differentiable functions are the ones for which $|\nabla f|$ is bounded above by L: it follows from their definition that for each $z \in D$ function f lies between two functions (see Figure 6):

$$f(z) - L||x-z|| \leq f(x) \leq f(z) + L||x-z||.$$

- β-smooth convex twice differentiable functions are the ones for which $\nabla^2 f$ is bounded above by β: it follows from 1. of Proposition 2.9 that for each $z \in D$ function f lies below the quadratic function with the main coefficient $\beta/2$, whose value and gradient at z coincides with f:

$$f(x) \leq f(z) + \nabla f(z)^T(x-z) + \frac{\beta}{2}||z-x||^2.$$

- α-strongly convex twice differentiable functions are the ones for which $\nabla^2 f$ is bounded below by α: it follows from 1. of Proposition 2.11 that for each $z \in D$ function f lies above the quadratic function with the main coefficient $\alpha/2$, whose value and gradient at z coincides with f:

$$f(x) \geq f(z) + \nabla f(z)^T(x-z) + \frac{\alpha}{2}||z-x||^2. \qquad (1)$$

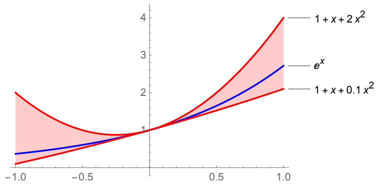

Figure 8: Function e^x on the interval $[-1,1]$ lies between parabolas with leading coefficients .1 and 2, which are tangent to the graph of e^x at $z=0$. Since the same bound holds for each $z \in [-1,1]$ (equivalently, since the second derivative of e^x on $[-1,1]$ lies on $[.2,4]$) we conclude e^x restricted to $[-1,1]$ is 4-smooth and .2-strongly convex.

[9] $\forall x, y : f(y) \geq f(x) + \nabla f(x)^T(y-x)$.

[10] If f is twice continuously differentiable then f is convex iff all eigenvalues of $\nabla^2 f$ are non-negative.

[11] Conditions below can often be achieved by restricting a function. Each continuously differentiable function on a closed bounded domain is Lipschitz. Each twice continuously differentiable function on a closed bounded domain is β-smooth and α-strongly convex.

[12] Convex twice differentiable functions are the ones for which $\nabla^2 f$ is bounded below by 0.

Observe that for $\alpha < \beta$ there are many convex β-smooth α-strongly convex functions with unbounded domain (for example, x^t for each $t \in [\alpha/2, \beta/2]$). However, an L-Lipschitz α-strongly convex function necessarily has bounded domain, see Figure 9.

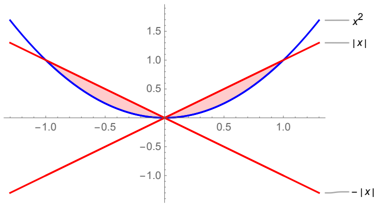

Figure 9: The domain of a 1-Lipschitz 1-strongly convex function f with $f(0) = f'(0) = 0$ is an interval contained in $[-1, 1]$ as its graph has to be contained in the shaded region.

A simple consequence of Equation (1)[13] is that the minimal value of f is above the minimal value of the corresponding quadratic[14], i.e.,

$$\min_{x \in D} f(x) \geq Q(x^*) = f(z) - \frac{1}{2\alpha}||\nabla f(z)||^2.$$

Why would we be interested on bounds on the second derivative? Let us consider case $n = 1$. In step k of GD we are using two values: $f(x_k)$ and $f'(x_k)$, which jointly determine the tangent to f at x_k. However, that does not tell us how large of a step we should make, i.e., what to choose for γ. From a geometric picture we can see that if $f''(x_k)$ is large[15], we should be making shorter steps than in the case when $f''(x_k)$ is small[16], see Figure 10. Parameters β and α will be playing a crucial role in determining our optimal step size γ.

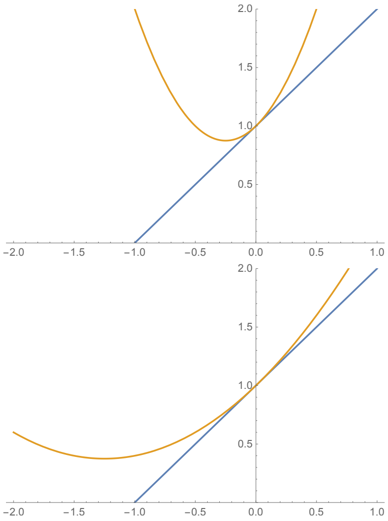

Figure 10: Two functions with $f(0) = f'(0) = 1$. Assume we are running a GD with $x_1 = 0$. The upper function has a higher second derivative which means we should be making a smaller step (smaller γ) that at the bottom function, whose second derivative is lower.

3 Guarantees for gradient descent

In this section we present a number of theoretical results[17] guaranteeing convergence of GD. In order to provide a condensed overview we state most of them in two theorems. In the first one we consider functions defined on $\mathbb{R}^n$.

> **Theorem 3.1.** *Assume $L, \alpha, \beta > 0, \alpha < \beta, f\colon \mathbb{R}^n \to \mathbb{R}$ is convex with global minimum at x^*, $x_1 \in \mathbb{R}^n$, and sequence x_k is obtained through GD, i.e., $x_{k+1} = x_k - \gamma \nabla f(x_k)$ for some $\gamma > 0$.*
>
> 1. *If f is L-Lipschitz and $\gamma = \frac{||x_1 - x^*||}{L\sqrt{T}}$ for some $T \in \mathbb{N}$, then*
>
> $$f\Big(\frac{1}{T}\sum_{i=1}^{T} x_i\Big) - f(x^*) \leq \frac{L||x_1 - x^*||}{\sqrt{T}}.$$
>
> 2. *If f is β-smooth and $\gamma = \beta^{-1}$, then*
>
> $$f(x_{k+1}) - f(x^*) \leq \frac{2\beta||x_1 - x^*||^2}{k}.$$
>
> 3. *If f is α-strongly convex and β-smooth, $\gamma = \frac{2}{\alpha+\beta}$ and $\kappa = \frac{\beta}{\alpha}$, then*
>
> $$f(x_{k+1}) - f(x^*) \leq \frac{\beta}{2}\Big(\frac{\kappa - 1}{\kappa + 1}\Big)^{2k}||x_1 - x^*||^2.$$

In the second theorem we state versions of results of Theorem 3.1 for functions defined on a closed convex set $D \subset \mathbb{R}^n$, i.e., $f\colon D \to \mathbb{R}$.

[13] Recall it holds for α-strongly convex twice differentiable functions.

[14] The quadratic being $Q(x) = f(z) + \nabla f(z)^T(x - z) + \frac{\alpha}{2}||z - x||^2$. Using differentiation it is easy to see its minimum is attained at $x^* = z - \frac{1}{\alpha}\nabla f(z)$

[15] Meaning the graph of f "curves" a lot.

[16] Meaning the graph of f "curves" just a little bit.

[17] The material presented here is extracted from Chapter 3 of Bubeck's book.
For functions f that are not strictly convex Theorem 3.1 does not imply $x_k \to x^*$.

For technical reasons we assume that a global minimum x^* is attained in the interior of D. We consider such functions for two reasons: first, our function may not be defined on the whole Euclidean space[18]; and second, by restricting our function to closed bounded subsets we typically attain our favourite properties[19] even though the original function itself does not posses[20] them. However, such restriction presents an issue: GD may return x_k outside our domain. A simple solution is the projected gradient descent: at each step we project x_k. Projection[21] is defined as $\pi_D\colon \mathbb{R}^n \to D$ and maps x to its closest point on D.

PROJECTED GRADIENT descent PGD:
$x_{k+1} = \pi_D(x_k - \gamma \nabla f(x_k))$

Theorem 3.3. *Assume $L, \alpha, \beta > 0, \alpha < \beta, f\colon D \to \mathbb{R}$ is a convex function with a global minimum x^* contained in the interior of the closed convex domain $D \subset \mathbb{R}^n$, $x_1 \in D$, and sequence x_k is obtained through PGD, i.e., $x_{k+1} = \pi_D(x_k - \gamma_k \nabla f(x_k))$ for some $\gamma_k > 0$.*

1. *If f is L-Lipschitz and $\gamma = \frac{||x_1 - x^*||}{L\sqrt{T}}$ for some $T \in \mathbb{N}$, then*

$$f\Big(\frac{1}{T}\sum_{i=1}^{T} x_i\Big) - f(x^*) \leq \frac{L||x_1 - x^*||}{\sqrt{T}}.$$

2. *If f is β-smooth and $\gamma = \beta^{-1}$, then*

$$f(x_k) - f(x^*) \leq \frac{3\beta||x_1 - x^*||^2 + f(x_1) - f(x^*)}{k}.$$

3. *If f is α-strongly convex and β-smooth, $\gamma = \frac{1}{\beta}$ and $\kappa = \frac{\beta}{\alpha}$, then*

$$f(x_{k+1}) - f(x^*) \leq \frac{\beta}{2}\Big(\frac{\kappa - 1}{\kappa}\Big)^{2k} ||x_1 - x^*||^2.$$

4. *If f is α-strongly convex and L-Lipschitz, and $\gamma_k = \frac{2}{\alpha(k+1)}$, then*

$$f\Big(\sum_{i=1}^{T} \frac{2i}{T(T+1)} x_i\Big) - f(x^*) \leq \frac{2L^2}{\alpha(T+1)}.$$

Note that we have not stated a result for α-strongly convex L-Lipschitz functions in the context of GD. The reason is, as we have already mentioned, that such functions always have bounded domain.

Lemma 3.2. *For the projection map $\pi_D\colon \mathbb{R}^n \to D$ to a closed convex domain D the following hold:*

1. *π_D is well defined, i.e., the closest point is unique.*
2. *$\forall x \in \mathbb{R}^n, \forall y \in D : ||x - y|| \geq ||\pi_D(x) - y||$.*

Proof. 1. If $x \in \mathbb{R}^n$ had two closest points $y, z \in D$, their midpoint would lie in D by convexity and would be even closer to X, a contradiction to the existence of two closest points.

2. If $||x - y|| < ||\pi_D(x) - y||$ then $||x - y|| \geq ||\pi_D(x) - x||$ implies the longest side A in the triangle $x, y, \pi_D(x)$ is between $\pi_D(x)$ and y, see Figure 11. By convexity $A \subseteq D$ and this A being the longest in the triangle, it contains a point $z \in A$, with $||z - x|| < ||x - \pi_D(x)||$, a contradiction with the definition of $\pi_D(x)$. □

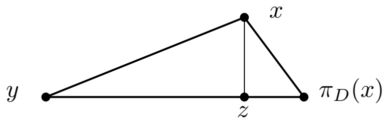

Figure 11: A sketch of the proof of 2. of Lemma 3.2.

The convergence rate for α-strongly convex L-Lipschitz function seems to be independent of the starting point. However, since the diameter of the domain of such a function is always bounded by some function $D(\alpha, L)$, the distance $||x_1 - x^*||$ is also bounded by $D(\alpha, L)$.

[18] Recall, for example, that the domain of a Lipschitz strongly convex function is always bounded.

[19] Being α-strongly convex, β-smooth, Lipschitz.

[20] For example think about e^x.

[21] See Lemma 3.2 on the side for some details. Finding such a projection is a challenging problem on its own. It is fairly straightforward though when D is a closed disc: we just use the radial projection.

Proofs

In this subsection we provide proofs of some of the stated results. The first proof is a model for many other convergence proofs for variants of GD.

Proof of 1. of Theorem 3.1. The proof has four distinct parts.

I: setting the basic equality. We use an elementary equality $||a-b||^2 = ||a||^2 + ||b||^2 - 2a^Tb$ in

$$\begin{aligned} ||x_{i+1} - x^*||^2 =& ||x_i - \gamma\nabla f(x_i) - x^*||^2 = \\ =& ||(x_i - x^*) - \gamma\nabla f(x_i)||^2 = \\ =& ||x_i - x^*||^2 + \gamma^2||\nabla f(x_i)||^2 - 2\gamma(x_i - x^*)^T\nabla f(x_i) \end{aligned}$$

to express

$$(x_i - x^*)^T\nabla f(x_i) = \frac{1}{2\gamma}(||x_i - x^*||^2 - ||x_{i+1} - x^*||^2) + \frac{\gamma}{2}||\nabla f(x_i)||^2.$$

II: applying properties of f. Since f is L-Lipschitz[22] and convex[23] we conclude

$$f(x_i) - f(x^*) \leq \frac{1}{2\gamma}(||x_i - x^*||^2 - ||x_{i+1} - x^*||^2) + \frac{\gamma L^2}{2}.$$

III: Telescoping sum. Adding the obtained inequalities

$$\begin{aligned} f(x_1) - f(x^*) &\leq \frac{1}{2\gamma}(||x_1 - x^*||^2 - ||x_2 - x^*||^2) + \frac{\gamma L^2}{2} \\ f(x_2) - f(x^*) &\leq \frac{1}{2\gamma}(||x_2 - x^*||^2 - ||x_3 - x^*||^2) + \frac{\gamma L^2}{2} \\ &\vdots \\ f(x_T) - f(x^*) &\leq \frac{1}{2\gamma}(||x_T - x^*||^2 - ||x_{T+1} - x^*||^2) + \frac{\gamma L^2}{2} \end{aligned}$$

most of the terms cancel out and we are left with

$$\begin{aligned} \sum_{i=1}^{T}(f(x_i) - f(x^*)) &\leq \frac{1}{2\gamma}(||x_1 - x^*||^2 - ||x_{T+1} - x^*||^2) + \frac{T\gamma L^2}{2} \\ &\leq \frac{1}{2\gamma}||x_1 - x^*||^2 + \frac{T\gamma L^2}{2} \quad \overset{\text{setting } \gamma}{=} \\ &= \sqrt{T}L||x_1 - x^*|| \end{aligned}$$

IV: Finishing touch. We divide the obtained inequality by T and since f is convex[24] We obtain

$$f\Big(\frac{1}{T}\sum_{k=1}^{T} x_i\Big) - f(x^*) \leq \frac{L||x_1 - x^*||}{\sqrt{T}}.$$

□

[22] Meaning $||\nabla f|| \leq L$.

[23] Meaning $(x_i - x^*)^T\nabla f(x_i) \geq f(x_i) - f(x^*)$.

[24] Meaning $\frac{1}{T}\sum_{i=1}^{T} f(x_i) \geq f(\sum_{i=1}^{T}\frac{1}{T}x_i)$.

In a similar fashion we can prove several other results, for example 1. of Theorem 3.3, which is a projected version of 1. of Theorem 3.1.

Proof of 1. of Theorem 3.3.

I: setting the basic equality. By 2. Lemma 3.2[25] we have

$$\begin{aligned}||x_{i+1} - x^*||^2 =&||\pi_D(x_i - \gamma\nabla f(x_i)) - x^*||^2 \leq \\ \leq&||x_i - \gamma\nabla f(x_i) - x^*||^2 = ...\end{aligned}$$

and then we continue through I-IV as in Proof of 1. of Theorem 3.1.

□

Proof of 4. of Theorem 3.3.

I: setting the basic equality. This part is the same as in the Proof of 1. of Theorem 3.3, we deduce

$$(x_i - x^*)^T\nabla f(x_i) = \frac{1}{2\gamma_i}(||x_i - x^*||^2 - ||x_{i+1} - x^*||^2) + \frac{\gamma_i}{2}||\nabla f(x_i)||^2.$$

II: applying properties of f. We incorporate the facts that f is α-strongly convex[26] and L-Lipschitz we obtain

$$f(x_i) - f(x^*) \leq (\frac{1}{2\gamma_i} - \frac{\alpha}{2})||x_i - x^*||^2 - \frac{1}{2\gamma_i}||x_{i+1} - x^*||^2 + \frac{\gamma_i L^2}{2},$$

and as $\gamma_i = \frac{2}{\alpha(i+1)}$ we conclude[27]

$$f(x_i) - f(x^*) \leq \frac{\alpha(i-1)}{4}||x_i - x^*||^2 - \frac{\alpha(i+1)}{4}||x_{i+1} - x^*||^2 + \frac{L^2}{\alpha(i+1)}.$$

III: Telescoping sum. In order to obtain a form suitable[28] for a telescopic sum we multiply the obtained inequality by index i to obtain

$$if(x_i) - if(x^*) \leq \frac{\alpha i(i-1)}{4}||x_i - x^*||^2 - \frac{\alpha i(i+1)}{4}||x_{i+1} - x^*||^2 + \frac{iL^2}{\alpha(i+1)}.$$

and perform the summation to obtain

$$\sum_{i=1}^{T}(if(x_i) - if(x^*)) \leq -\frac{\alpha T(T+1)}{4}||x_{T+1} - x^*||^2 + \sum_{i=1}^{T}\Big(\frac{iL^2}{\alpha(i+1)}\Big).$$

This simplifies[29] to

$$\sum_{i=1}^{T} if(x_i) - \frac{T(T+1)}{2}f(x^*) \leq \frac{TL^2}{\alpha}.$$

IV: Finishing touch. Dividing the inequality by $\frac{T(T+1)}{2}$ and using convexity[30] we obtain

$$f\Big(\sum_{i=1}^{T}\frac{2i}{T(T+1)}x_i\Big) - f(x^*) \leq \frac{2L^2}{\alpha(T+1)}.$$

□

[25] Meaning $||\pi_D(x_i - \gamma\nabla f(x_i)) - x^*|| \leq ||(x_i - \gamma\nabla f(x_i)) - x^*||$.

[26] Meaning $f(x_i) - f(x^*) \leq -\frac{\alpha}{2}||x_i - x^*||^2 + (x_i - x^*)^T\nabla f(x_i)$.

[27] Note that $\frac{1}{2\gamma_i} - \frac{\alpha}{2} = \frac{\alpha(i+1)}{4} - \frac{2\alpha}{4} = \frac{\alpha(i-1)}{4}$.

[28] Meaning where most terms will cancel out.

[29] As $\sum_{i=1}^{T} i = \frac{T(T+1)}{2}$, $\frac{i}{i+1} \leq 1$, and $-\frac{\alpha T(T+1)}{4}||x_{T+1} - x^*||^2 < 0$.

[30] Note that $\sum_{i=1}^{T}\frac{2i}{T(T+1)} = 1$.

Proof of 3. of Theorem 3.1. The first two parts follow the previous proofs. **I: setting the basic equality**: As $\nabla f(x^*) = 0$ we can express it as

$$(x_i - x^*)^T(\nabla f(x_i) - \nabla f(x^*)) = \\ = \frac{1}{2\gamma}(||x_i - x^*||^2 - ||x_{i+1} - x^*||^2) + \frac{\gamma}{2}||\nabla f(x_i)||^2.$$

Lemma 3.4. *In the context of 3. of Theorem 3.1 the following holds: $\forall x, y \in \mathbb{R}^n$, one has*

$$(\nabla f(x) - \nabla f(y))^T(x-y) \geq \\ \frac{\alpha\beta||x-y||^2}{\alpha+\beta} + \frac{||\nabla f(x) - \nabla f(y)||^2}{\alpha+\beta}.$$

This lemma is proved in Bubeck's book as Lemma 3.11.

II: applying properties of f. By Lemma 3.4 we obtain

$$\frac{\alpha\beta||x_i - x^*||^2}{\alpha+\beta} + \frac{||\nabla f(x_i) - \nabla f(x^*)||^2}{\alpha+\beta} \leq \\ \leq \frac{1}{2\gamma}(||x_i - x^*||^2 - ||x_{i+1} - x^*||^2) + \frac{\gamma}{2}||\nabla f(x_i)||^2.$$

which we can multiply by 2γ, and use $\nabla f(x^*) = 0$ again to rearrange it into

$$||x_{i+1} - x^*||^2 \leq \Big(1 - \frac{2\alpha\beta\gamma}{\alpha+\beta}\Big)||x_i - x^*||^2 - \Big(\frac{-2\gamma}{\alpha+\beta} + \gamma^2\Big)||\nabla f(x_i||^2).$$

Since $\gamma = \frac{2}{\alpha+\beta}$ we get[31]

$$||x_{i+1} - x^*||^2 \leq \Big(\frac{\kappa-1}{\kappa+1}\Big)^2 ||x_i - x^*||^2$$

and hence

$$||x_{i+1} - x^*||^2 \leq \Big(\frac{\kappa-1}{\kappa+1}\Big)^{2i} ||x_1 - x^*||^2.$$

The proof is concluded using Proposition 2.9 which states[32]

$$f(x_{i+1}) - f(x^*) \leq \frac{\beta}{2}||x_{i+1} - x^*||^2.$$

□

Summary

Table 1 summarizes presented convergence rates[33]. Recall T is the number of steps, $\kappa = \frac{\beta}{\alpha}$. Positive number R denotes $||x_1 - x^*||$ for GD and the diameter of D for PGD.

How much can we improve these bounds? Later we will present a number of variants of GD which try to improve these convergence results, sometimes for a specific class of functions. In the next section however we will present a lower bound, from which it follows that in a way some of the bounds obtained above are optimal.

4 Lower bounds for black box procedures

Lower bounds provide an estimate about how well GD or any so called black box model can perform for a given class of functions. Let us explain this statement.

[31] Using $1 - \frac{2\alpha\beta\gamma}{\alpha+\beta} = \left(\frac{\kappa-1}{\kappa+1}\right)^2$ and $\frac{-2\gamma}{\alpha+\beta} + \gamma^2 = 0$.

[32] Here we use the facts that $\nabla f(x^*) = 0$ and $f(x^*) \leq f(x_{i+1})$.

[33] It contains the reference numbers of the corresponding theorems in Bubeck's book.

	GD	PGD
L-Lipschitz	$\frac{RL}{\sqrt{T}}$ [3.1]	$\frac{RL}{\sqrt{T}}$ [3.1]
β-smooth	$\frac{2\beta R^2}{T-1}$ [3.3]	$\frac{3\beta R^2+f(x_1)-f(X^*)}{T}$ [3.7]
α-strongly convex, β-smooth	$\frac{\beta}{2}\left(\frac{\kappa-1}{\kappa+1}\right)^{2(T-1)}R^2$ [3.12]	$\frac{\beta}{2}\left(\frac{\kappa-1}{\kappa}\right)^{2(T-1)}R^2$ [3.10]
α-strongly convex, L-Lipschitz	NA, domain always bounded	$\frac{2L^2}{\alpha(T+1)}$ [3.9]

Table 1: Convergence rates for GD and PGD with T being the number of executed steps and $R = ||x_1 - x^*||$. Added are reference numbers of the corresponding theorems in Bubeck's book.

A black box model assumes an iterative search for a minimum of a function using information about all past gradients and all past steps. GD is an example of a black box model. Here is how it works. Given a function f we choose $x_1 \in D$. For the sake of simplicity[34] we assume $x_1 = 0$. We then iteratively choose x_k so that

$$x_k \in \mathrm{Span}\{\nabla f(x_1), \nabla f(x_2), \ldots, \nabla f(x_{k-1})\}. \tag{2}$$

In particular, each our step is a linear combination of gradients in previous points. Obviously GD is a black box model while PGD is not.

It turns out we can show that a black box model can perform only[35] so well for x_k with $k < n$. Below we present one formal statements of such an idea[36]. Here is a simple primer on the idea of the proofs. Consider a quadratic function $f(x, y) = 3x^2 + y^2$ (for example, we can look at quadratic on Figure 1). Starting with a point x_1 outside the main axis, the subspace $x_1 + t\nabla f(x_1), t \in \mathbb{R}$ will not contain minimum $(0, 0)$ hence we can't get to it using a black box procedure in one step.

Theorem 4.1. *Assume $n = 2m+1, \beta > 0$. There exists a β-smooth strictly convex function f such that for any black box procedure*

$$\min_{1\le s\le m} \Big(f(x_s) - f(x^*)\Big) \ge \frac{3\beta}{32}\frac{||x_1 - x^*||^2}{(m+1)^2}.$$

Proof. We first define a quadratic function f. Start with a tridiagonal

[34] In general a black box model assumes an iterative search for a minimum of a function so that each step is a linear combination of all past gradients and steps, i.e., $x_k - x_{k-1}$ is contained in

$$\mathrm{Span}\Big\{\nabla f(x_1), \ldots, \nabla f(x_{k-1}),$$
$$x_2 - x_1, \ldots, x_{k-1} - x_{k-2}\Big\}.$$

If $x_1 = 0$ then all previous steps are also contained in the span of previous gradients and this condition simplifies to (2).

[35] This means that even if we can optimize our choice of parameters γ_k in a GD specifically for our function, we can only do so well. Of course, when $k = n$ we might generically expect the past gradients to span $\mathbb{R}^n$ and hence we can get to a solution in one step.

[36] This kind of results are are typically proved by providing a counterexample. For a few more results of this sort see Theorems 3.13 and 3.15 in Bubeck's book.

matrix

$$k \text{ rows} \left\{ \overbrace{\begin{pmatrix} 2 & -1 & 0 & 0 & \cdot & \cdot & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdot & \cdot & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & -1 & 2 & -1 \\ 0 & 0 & 0 & \cdot & \cdot & 0 & -1 & 2 \end{pmatrix}}^{k \text{ columns}} \right. = A_k.$$

Setting $A_n = A$ and e_i to be the standard i^{th} basis vector we define

$$f(x) = \frac{\beta}{8} x^T A x - \frac{\beta}{4} x^T e_1.$$

Note[37] that $\nabla f(x) = \frac{\beta}{4} Ax - \frac{\beta}{4} e_1$ and $\nabla^2 f(x) = \frac{\beta}{4} A$, and since the eigenvalues of A are between 0 and 4 by Lemma 4.2 the second order conditions imply f is strictly convex and β-smooth.

Lemma 4.2. *Eigenvalues of A_k lie on $(0,4]$.*

Proof. Let $z = (z_1, z_2, \dots, z_k)$ be an eigenvector of A with eigenvalue ℓ. Note that $z^T A z = \ell z^T z = \ell ||z||^2$, and also $z^T A z = 2\sum_{i=1}^{k} z_i^2 - 2\sum_{i=1}^{k-1} z_i z_{i+1} = z_1^2 + z_k^2 + \sum_{i=1}^{k-1}(z_i - z_{i+1})^2 \geq 0$, with the inequality being strict unless $x_i = 0, \forall i$. Hence $\ell \geq 0$.

On the other hand we can use 1-norm $|z|_1 = \sum_{i=1}^{k} |z_i|$ and the fact that the sum ob absolute values in each column (which is the 1-norm of that column) is at most 4 to deduce $|Az|_1 \leq \sum_{i=1}^{k} 4|z_i| = |4z|_1$, hence $\ell \leq 4$. □

We next think about subspaces that could contain steps x_i in a black box model.

- $x_1 = 0$.
- $\nabla f(x_1) = -\frac{\beta}{4} e_1$, so $x_2 \in \text{Span}\{e_1\}$.
- $\nabla f(x_2) = \frac{\beta}{4} A x_2 - \frac{\beta}{4} e_1$, which is contained in $\text{Span}\{e_1, e_2\}$ as A is tridiagonal, and so $x_3 \in \text{Span}\{e_1, e_2\}$.
- We inductively deduce that $x_i \in \text{Span}\{e_1, e_2, \dots, e_{i-1}\}$.

So the question is what is the minimum of f_k, which is defined as the restriction[38] of f to $\text{Span}\{e_1, e_2, \dots, e_k\}$. The minimum x_k^* of f_k satisfies $\nabla f(x_k^*) = 0$ which[39] is a system of linear equations $A_k x_k^* = e_1$, whose solution can be verified to be

$$x_k^* = (1 - \frac{1}{k+1}, 1 - \frac{2}{k+1}, \dots, 1 - \frac{k}{k+1}). \tag{3}$$

By strict convexity the minimum of f_k is unique. Thus[40]

$$||x_k^*||^2 = \sum_{i=1}^{k} \Big(\frac{k-i+1}{k+1}\Big)^2 = \sum_{j=1}^{k} \Big(\frac{j}{k+1}\Big)^2 = \frac{k(k+1)(2k+1)}{6(k+1)^2} < \frac{k+1}{3}$$

and[41]

$$f_k(x_k^*) = \frac{\beta}{8} x_k^{*T} A_k x_k^* - \frac{\beta}{4} x_k^{*T} e_1 = -\frac{\beta}{8} x_k^{*T} e_1 = -\frac{\beta}{8}\Big(1 - \frac{1}{k+1}\Big).$$

We can now conclude[42] by

$$f(x_s^*) - f(x^*) \geq f(x_m^*) - f(x^*) = \frac{\beta}{8}\Big(\frac{1}{m+1} - \frac{1}{n+1}\Big) >$$

$$\frac{\beta}{16(m+1)} = \frac{3\beta\frac{m+1}{3}}{16(m+1)^2} = \frac{3\beta\frac{2m+2}{3}}{32(m+1)^2} > \frac{3\beta||x_{2m+1}^*||^2}{32(m+1)^2} = \frac{3\beta||x^*||^2}{32(m+1)^2}.$$

□

[37] Recall $\frac{\partial x^T A x}{\partial x} = (A + A^T)x$.

[38] In this case

$$f_k(x) = \frac{\beta}{8} x^T A_k x - \frac{\beta}{4} x^T e_1.$$

[39] Recall $\nabla f_k(x) = \frac{\beta}{4} A_k x - \frac{\beta}{4} e_1$.

[40] Taking into account equalities $1 - \frac{i}{k+1} = \frac{k-i+1}{k+1}$, $j = k - i + 1$, and $\sum_{j=1}^{k} j^2 = \frac{k(k+1)(2k+1)}{6}$.

[41] Using $A_k x_k^* = e_1$ from above.

[42] Using $n + 1 = 2m + 2$.

5 Modifications of gradient descent

In this section we present a number of modifications of GD. We start by analysing two-dimensional quadratic functions, which will explain some of our theoretical results.

Quadratics

We start with the one-dimensional case: $f(x) = ax^2$ for $a > 0$. We know it's minimum is at 0. In an analogous way to Example 1.1 it is easy to see that GD with learning rate γ results in

$$x_{k+1} = (1 - 2\gamma a)^k x_1,$$

so the convergence rate will depend on $|1 - 2\gamma a|$.

A bit more challenging is a general quadratic case. Let H be symmetric positive definite and define $f(x) = \frac{1}{2}x^T Hx$. As $\nabla f(x) = Hx$ we can deduce

$$x_{k+1} = x_k - \gamma Hx_k = (I - \gamma H)x_k = (I - \gamma H)^k x_1.$$

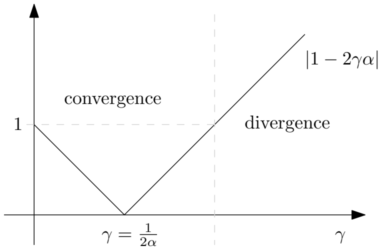

Figure 12: Convergence of one-dimensional quadratic: optimal choice of γ is $\frac{1}{2\alpha}$, at which GD converges in one step. For $\gamma > \frac{1}{\alpha}$ GD diverges.

If we define $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ as the eigenvalues of H we have[43] $\lambda_1||x|| \leq ||Hx|| \leq \lambda_n||x||$. Define[44] $\alpha = \lambda_1$ and $\beta = \lambda_n$. By the triangle inequality[45] we deduce

$$||1 - \gamma H|| = \max\{|1 - \alpha\gamma|, |1 - \beta\gamma|\}.$$

We now want to choose the learning rate γ which will minimize $||1 - \gamma H||$ as this will optimize the convergence rate. From Figure 13 we calculate that this happens at $\gamma = \frac{2}{\alpha+\beta}$. We make two observations:

- If at the one-dimensional case the optimal γ was obtained by dividing by the second derivative 2α representing the curvature, in a general case we divide by the average between the largest and the smallest curvatures, i.e., the average between the largest and the smallest "directional second derivatives". This is actually the learning rate for a α-strongly convex and β-smooth function in 3. of Theorem 3.1.
- Having chosen $\gamma = \frac{2}{\alpha+\beta}$, we see that for $\kappa = \frac{\beta}{\alpha}$ we have

  $$\max\{|1 - \alpha\gamma|, |1 - \beta\gamma|\} = 1 - \alpha\frac{2}{\alpha + \beta} = \frac{\kappa - 1}{\kappa + 1},$$

  which again appears in 3. of Theorem 3.1.

So as far as GD with a constant learning rate γ is concerned, 3. of Theorem 3.1 provides optimal bounds. However, improved convergence may be obtained by Polyak version of GD.

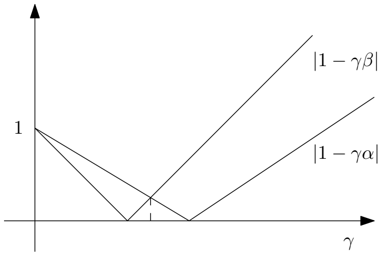

Figure 13: Convergence of a quadratic: optimal choice of γ is the solution of $1 - \gamma\alpha = \gamma\beta - 1$, which is at $\gamma = \frac{2}{\alpha+\beta}$.

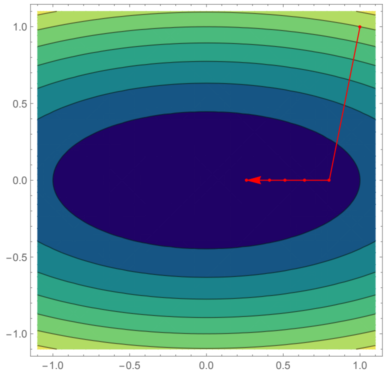

Figure 14: Convergence on a quadratic: for $f(x,y) = x^2 + 5y^2$ and $\gamma = \frac{1}{10}$ we obtain the following sequence. The behaviour can be explained using the discussed convergence analysis as $|1 - \gamma\beta| = 0$.

[43] Since H is symmetric positive definite, its eigenvalues are positive and coincide with its singular values.

[44] f is λ_1-strongly convex and λ_2-smooth.

[45] Alternatively, we can observe that $1 - \gamma H$ is symmetric with eigenvalues $1 - \gamma\lambda_1, \ldots, 1 - \gamma\lambda_n$.

Polyak (also called momentum or heavy ball) GD

Polyak's method from 1966 introduces a momentum term in the form of $\mu(x_k - x_{k-1})$.

POLYAK GD:
$$x_{k+1} = x_k - \gamma \nabla f(x_k) + \mu(x_k - x_{k-1})$$

This modification[46] makes x_{k+1} dependent on $\nabla(x_k)$ and on the previous step $x_k - x_{k-1}$. The momentum term represents a momentum a heavy ball would be carrying into next steps while rolling down along the graph of f.

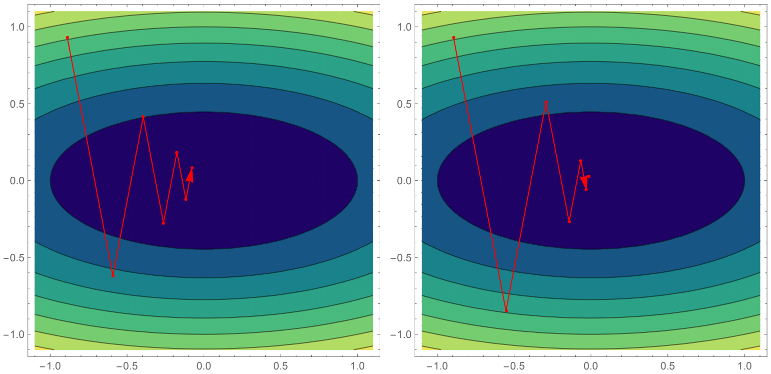

Figure 15: Optimal convergence via GD (left, $\gamma = 1/6$) and via Polyak GD (right, $\gamma \approx .19$, $\mu \approx .15$) for function $x^2 + 5y^2$, whose minimal value is 0. The value at x_6 of GD is .04 while Polyak gives 0.004, which is much closer to 0, even though x_2 of Polyak is clearly worse.

Using Figure 15 we can get some intuition about why Polyak GD works well. Going back to the analysis of quadratic functions note that we choose $\gamma = \frac{2}{\alpha+\beta}$ so that $\max\{|1-\alpha\gamma|, |1-\beta\gamma|\}$ is minimal, which means $1 - \alpha\gamma > 0$ and $1 - \beta\gamma < 0$. Now $1 - \alpha\gamma > 0$ implies that the corresponding coordinate[47] decreases monotonously[48], which we can observe on Figure 15. Inequality $1 - \beta\gamma < 0$ however implies that the corresponding coordinate[49] is alternating [50] The momentum term consequently absorbs some of this alternation and turns it into faster convergence.

We proceed by convergence analysis for Polyak GD for quadratic function. Let H be symmetric positive definite with eigenvalues on $[\alpha, \beta]$ and define $f(x) = \frac{1}{2}x^T Hx$ with the obvious minimal value 0 at $(0, 0, \ldots, 0)$. We encode the iterative step in the following way:

$$\begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix} = \overbrace{\begin{bmatrix} I - \gamma H + \mu I & -\mu I \\ I & 0 \end{bmatrix}}^{A} \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix}$$

Just like in the case of GD for quadratics we want to estimate $||A^k||$ because $||x_{k+1}|| \leq ||A^k|| \cdot ||x_1||$. Unfortunately, as A is not symmetric $||A||$ may not coincide with λ, which is defined[51] as the

[46] By default we assume that in the first step, when computing x_2, we only perform the standard GD as there is no previous step yet. Equivalently, we can define $x_0 = x_1$ and proceed with the Polyak iteration throughout all indices.

[47] On Figure 15 that is the x-coordinate as $\alpha = 2$ corresponds to it.

[48] See Example 1.1 for the same phenomenon.

[49] On Figure 15 that is the y-coordinate as $\beta = 10$ corresponds to it.

[50] Again, see Example 1.1 for the same phenomenon.

[51] Let us also define α as the minimal eigenvalue of H and β as the maximal eigenvalue of H.

maximal absolute value of an eigenvalue of A. However, by Lemma 5.1 it suffices to estimate $||A^k||$ asymptotically. So let's start looking for λ.

Lemma 5.1. *[A matrix analysis lemma] There exist $\varepsilon_k \to 0$ such that $||A^k|| \leq (\lambda + \varepsilon_k)^k$.*

Since H is diagonalizable in an orthonormal basis, we can use the same basis to transform A into a block diagonal matrix A' consisting of blocks

$$A_i = \begin{bmatrix} 1 - \gamma\lambda_i + \mu & -\mu \\ 1 & 0 \end{bmatrix},$$

with λ_i being the i^{th} eigenvalue of H. This means that the eigenvalues of A coincide with the list of eigenvalues of all A_i and in particular, λ is the largest absolute value of an eigenvalue of A_i. So let's find the eigenvalues of A_i and keep in mind that we want[52] their absolute value to be as small as possible[53] in order to get the best possible convergence.

First note that $|A_i| = \mu$ meaning[54] that in the optimal case both eigenvalues will be of absolute value $\sqrt{\mu}$. This happens if the eigenvalues are either both the same ($\sqrt{\mu}$ or $-\sqrt{\mu}$) or complex conjugates of each other. These two conditions are equivalent to the discriminant D of the characteristic polynomial of A_i being non-positive[55]. So let's find conditions on μ and γ that imply $D \leq 0$.

From A_i we conclude that $D = (1 - \gamma\lambda_i + \mu)^2 - 4\mu$. Taking into account[56] $\gamma \in [0,1]$ we can compute that $D \leq 0$ iff[57] $|1 - \sqrt{\gamma\lambda_i}| \leq \sqrt{\mu}$. This condition has to hold for all i so we must choose μ and γ so that

$$\sqrt{\mu} \geq \max_i\{|1 - \sqrt{\gamma\lambda_i}|\} = \max\{|1 - \sqrt{\gamma\alpha}|, |1 - \sqrt{\gamma\beta}|\}.$$

Similarly as in the case of quadratics we can use Figure 16 to see that as a function of γ the expression $\max\{|1 - \sqrt{\gamma\alpha}|, |1 - \sqrt{\gamma\beta}|\}$ is smallest at $\gamma = \frac{4}{(\sqrt{\alpha}+\sqrt{\beta})^2}$ and $\sqrt{\mu} = \frac{\sqrt{\beta}-\sqrt{\alpha}}{\sqrt{\beta}+\sqrt{\alpha}} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} < 1$. We thus proved a theorem, which implies that Polyak GD performs better than GD for quadratics.

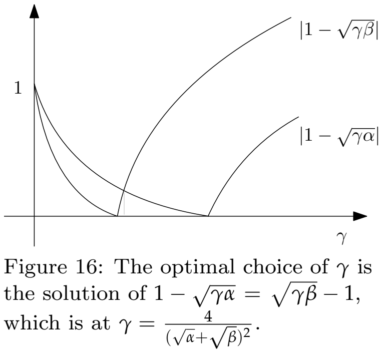

Figure 16: The optimal choice of γ is the solution of $1 - \sqrt{\gamma\alpha} = \sqrt{\gamma\beta} - 1$, which is at $\gamma = \frac{4}{(\sqrt{\alpha}+\sqrt{\beta})^2}$.

Theorem 5.2. *For a quadratic function the Polyak GD with parameters*

$$\sqrt{\mu} = \frac{\sqrt{\beta} - \sqrt{\alpha}}{\sqrt{\beta} + \sqrt{\alpha}}, \quad \gamma = \frac{4}{(\sqrt{\alpha} + \sqrt{\beta})^2}$$

satisfies

$$||x_{k+1} - x^*|| \leq \Big(\frac{\sqrt{\beta} - \sqrt{\alpha}}{\sqrt{\beta} + \sqrt{\alpha}} + \varepsilon_k\Big)^k ||x_1 - x^*||$$

For the conclusion we discuss some properties of Polyak GD:

[52] Meaning we want to make the appropriate choice of γ and μ.

[53] ...and in any case smaller than 1...

[54] Since the determinant equals the product of the eigenvalues.

[55] If $D > 0$ then one of the eigenvalues would be larger that $\sqrt{\mu}$, which is a less favourable situation.

[56] We can later check that our choice of γ satisfies this condition.

[57] $D \leq 0$ is equivalent to a system of two inequalities:

$$-2\sqrt{\mu} \leq 1 - \gamma\lambda_i + \mu \leq 2\sqrt{\mu}.$$

The left one is equivalent to

$$\gamma\lambda_i \leq 1 + 2\sqrt{\mu} + \mu = (1 + \sqrt{\mu})^2$$

and expressing $\sqrt{\mu}$ we get $\sqrt{\gamma\lambda_i} - 1 \leq \sqrt{\mu}$. The right one similarly reduces to $1 - \sqrt{\gamma\lambda_i} \leq \sqrt{\mu}$. Jointly we get $|1 - \sqrt{\gamma\lambda_i}| \leq \sqrt{\mu}$.

- Theorem 3.15 in Bubeck's book, which is another lower bound result for a black box method[58], implies that asymptotically speaking this convergence is as good as it can possibly be.

- We next mention a curiosity about Polyak GD. In the theorem above we picked specific optimal values for parameters μ and γ. Now assume we chose $\sqrt{\mu} \in \left(\frac{\sqrt{\beta}-\sqrt{\alpha}}{\sqrt{\beta}+\sqrt{\alpha}}, 1\right)$. In this case, judging by Figure 17, we have more flexibility for the choice of γ, it can be anywhere along the bold line. The argument above still works, the determinants of A_i are still μ and hence

  $$||x_{k+1} - x^*|| \leq \left(\sqrt{\mu} + \varepsilon_k\right)^k ||x_1 - x^*||.$$

  Surprisingly, the convergence rate depends only on μ and not on γ. This means that the choice of the momentum coefficient μ has more influence to the convergence than the gradient coefficient γ.

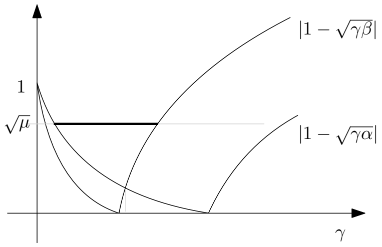

Figure 17: For $\mu < 1$ above the optimal value we can choose γ anywhere along the bold line and still satisfy $\sqrt{\mu} \geq \max\{|1-\sqrt{\gamma\alpha}|, |1-\sqrt{\gamma\beta}|\}$.

- Polyak GD has guaranteed great performance on quadratics and is also frequently used for general functions with good results. However, it turns out there are fairly simple examples of functions where it does not converge. For example, let $f\colon \mathbb{R} \to \mathbb{R}$ be defined by

  $$f'(x) = \begin{cases} 25x & x < 1 \\ x + 24 & 1 \leq x \leq 2 \\ 25x - 24 & 2 < x \end{cases}$$

  This means f is 25-smooth, 1-strongly convex[59] and consists of three parabolic segments. If we choose optimal parameters $\mu = \frac{4}{9}$ and $\gamma = \frac{1}{9}$ the following turns out to be true:

  - There exist points $w_1 \approx .65, w_2 \approx -1.8, w_3 \approx 2.12 \in \mathbb{R}$ so that if $x_1 = w_1$ and $x_2 = w_2$, then $x_3 = w_3, x_4 = w_1, x_5 = w_2, \ldots$, meaning that starting with such prescribed x_1 and x_2, the Polyak GD does not converge but loops indefinitely through values w_1, w_2, w_3.
  - The previous case is a bit artificial since in practice we typically define[60] $x_1 = x_2$. However, it can be proved that even if we start with $x_1 = x_2 \approx 3.3$, the terms x_i will start approaching the cyclic orbit w_1, w_2, w_3. In particular, x_i does not converge.

Nesterov GD

Nesterov GD is a slight modification of Polyak GD. While Polyak GD makes the gradient step first and then adds the momentum, Nesterov GD can be thought of as adding the momentum first and then making the gradient step from the new point.

[58] It is easy to see that Polyak GD is a black box method.

[59] With both these paramaters being optimal.

[60] A fun fact about polyak GD: assume $x_1 = x_2 \neq x^*$. Then even if $x_k = x^*$ for some k, we still have $x_{k+1} \neq x^*$ as the momentum term perturbs the next step.

NESTEROV GD:
$x_{k+1} = x_k - \gamma \nabla f\big(x_k + \mu(x_k - x_{k-1})\big) + \mu(x_k - x_{k-1})$

Theorem 5.3 (Theorem 3.18 in Bubeck's book)**.** *If f is α-strongly convex and β-smooth, $\gamma = \frac{1}{\beta}, \mu = \frac{\sqrt{\kappa}-1}{\sqrt{k}+1}$ and $\kappa = \frac{\beta}{\alpha}$, then*

$$f(x_{k+1}) - f(x^*) \le \frac{\alpha+\beta}{2}\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}}\Big)^k ||x_1 - x^*||^2.$$

Theorem 5.4 (Theorem 3.19 in Bubeck's book)**.** *If f is convex and β-smooth, and $\gamma = \frac{1}{\beta}$, then for an appropriate choice of μ_i we have*

$$f(x_k) - f(x^*) \le \frac{2\beta}{k^2}||x_1 - x^*||^2.$$

Nesterov GD provably works for more general functions than Polyak GD although the convergence for quadratics is slightly worse than that of Polyak GD[61].

Stochastic GD

Stochastic GD (SGD) becomes useful when our function f is a sum of a large number of functions[62], i.e., when $f = \frac{1}{N}\sum_{i=1}^{N} f_i$ and N is large.

Following the GD, each step would require us to:

1. Compute N gradients ∇f_i and add them up.
2. Evaluate at the current point and make the step.

In order to ease on the number of gradient computations we may do the following:

1. Compute the gradient ∇f_j for some random j.
2. Evaluate at the current point and make the step.

This way we reduced the number of gradient computations but typically also reduced the quality of our step. In particular, as soon as we computed one gradient we used it to make a step, without waiting for the other gradients to be computed. This is the idea of SGD.

[61] For a detailed comparison see L. Lessard, B. Recht, A. Packard: *Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints*, arXiv:1408.3595, especially the comparison table appearing on page 6.

[62] Imagine fitting a linear function (i.e., linear regression) to N points on the plane via the least squares method. The error function that should be minimized to obtain such a linear function is the sum of all squares of errors, i.e., it consists of N summands.

SGD:
$\forall k$ *choose* $j(k) \in \{1, 2, \ldots, N\}$ *randomly uniformly*
$x_{k+1} = x_k - \gamma \nabla f_{j(k)}(x_k)$

Theoretical guarantees on convergence can be deduced from the results for GD because $E[\nabla f_{j(k)}] = \nabla f$. For example, under the conditions of 1. of Theorem 3.1 we can conclude that SGD results in

$$E\Big[f\Big(\frac{1}{T}\sum_{i=1}^{T} x_i\Big)\Big] - f(x^*) \leq \frac{L||x_1 - x^*||}{\sqrt{T}}.$$

A downside of SGD is that after a while[63] the steps tend to bounce around the optimum[64]. To avoid it we typically use decaying step size γ_i. Another option that tackles this issue to a point is to use the minibatch[65] SGD. The idea is that instead of using just one gradient ∇f_i in each step, we use T such gradients with $T << N$. This is a variant between GD and SGD and which sometimes offers a good balance between the two.

MINIBATCH SGD:
$\forall k$ *choose* $j(k,1), \ldots, j(k,T) \in \{1, \ldots, N\}$ *randomly uniformly*
$x_{k+1} = x_k - \gamma \frac{1}{T} \sum_{i=1}^{T} \nabla f_{j(k,i)}(x_k)$

AdaGrad GD

Recall the standard GD:

$$x_{k+1} = x_k - \gamma \nabla f(x_k).$$

In the effort to improve GD, we can add a preconditioner[66] matrix A to obtain:

$$x_{k+1} = x_k - \gamma A \nabla f(x_k).$$

This modification will change the direction of our step[67]. However, we get added flexibility. While scalar γ can be used to weight the length of iterative steps, γA can be used to weight each coordinate of the gradient independently. In particular, if A is diagonal, then its i^{th} diagonal entry will be a weight on the i^{th} coordinate of the gradient step. In this subsection let $x^{(i)}$ denote the i^{th} component of a vector x.

AdaGrad (Adaptive GD) adapts to changes in each component of the gradient descent independently. For each i, k define $\tilde{d}_{i,k} = \sum_{j=1}^{k} \Big((\nabla f x_j)^{(i)} \Big)^2$ to be the sum of the squares of the i^{th} components of all the gradients up to step k and define[68] a diagonal matrix

[63] After most variables are well tuned.

[64] This is called the noise ball effect

[65] In this setting the classical GD is sometimes referred to as batch GD.

[66] Another way of thinking about it is that we replace parameter γ by a matrix γA. As will be explained in the next section, the optimal preconditioner is the inverse of the Hessian matrix. In this section however we are considering first order methods and therefore we will focus on preconditioners derived from gradients. AdaGrad fits into the description of quasi-Newton methods, which will be introduced in the following section.

[67] Meaning that we may not get a black box method anymore

[68] In order to avoid division by 0, a small positive value is often added to each diagonal element of D_1.

$$D_k = Diag[(\tilde{d}_{1,k})^{-1/2}, (\tilde{d}_{2,k})^{-1/2}, \ldots, (\tilde{d}_{n,k})^{-1/2}].$$

AdaGrad:

$$x_{k+1} = x_k - \gamma D_k \cdot \nabla f(x_k)$$

In particular, the i^{th} component of the k^{th} gradient step gets divided by $\sqrt{\tilde{d}_{i,k}}$, with $\tilde{d}_{i,k}$ being the cumulative sum of the squares of all i^{th} components of past gradients and hence contains the process history.

- Multiplication by D is accelerating the process in those components, whose gradient component is typically small[69]. On the other hand it is dampening the process in those directions that typically have large gradient component.
- Consequently AdaGrad promotes less visible features, and dampens updates in coordinates that oscillate a lot.
- AdaGrad performs well for sparse data.
- The problem is that the accumulating weights $\tilde{d}_{i,k}$ are an increasing function and eventually dampen the whole process. Thus while AdaGrad may perform well initially, it may get eventually stuck.

The shortcomings of AdaGrad are addressed by a number of other methods: RMSProp (which changes the definition of $\tilde{d}_{i,k}$ by putting more emphasis on the resent history instead of the uniformly treatment by AdaGrad), Adam[70] (it employs a variant of the modification of RMSProp but also changes the gradient term by incorporating past gradients and momenta), etc.

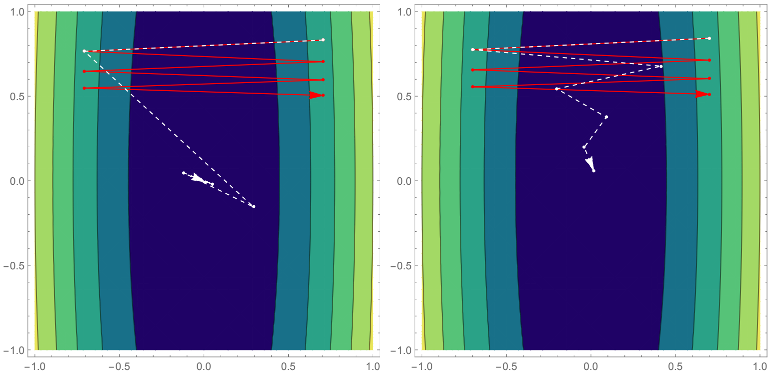

Figure 18: A comparison of GD (red) and AdaGrad (dashed white) on the left, and of GD (red) and RMSProp (dashed white) on the right. We see that the initial two points (i.e., the initial point and the initial step) are the same in all cases. Afterwards the weights on AdaGrad and RMSProp start having effect by dampening the oscillation in the x direction, which is still present in GD, and by promoting the small but consistent changes in the y direction.

[69] In terms of the absolute value, of course.

[70] It is actually a generalization of both the Polyak method and AdaGrad.

Summary

We summarize the variants of GD presented in this section[71]:

1. Polyak GD:

   + Works provably great for quadratics.
   + In practice often works well in general.
   - May not converge in general.

2. Nesterov GD:

   + Provably works well in general.
   + Still works well for quadratics, although slightly slower than Polyak.

3. SGD:

   + Accelerates convergence for functions of the form $f = \sum_{i=1}^{N} f_i$ when N is large.
   + Each iterative step is much cheaper than at GD.
   - More steps are needed.
   - Eventual variance may make our steps bounce around.

4. Minibatch SGD:

   + Accelerates convergence for functions of the form $f = \sum_{i=1}^{N} f_i$ when N is large, although less than SGD.
   + Each iterative step is cheaper than at GD.
   - More steps are needed, although less than with SGD.
   - Eventual variance may make our steps bounce around, although less than SGD.

5. AdaGrad GD:

   + Initially works great.
   - May eventually get stuck due to accumulating weights.

6 Newton and quasi-Newton methods

Newton method[72] is second order method, which means we will be making inductive guessing[73] based on the second derivative as well as the first derivative. Subsequently derived quasi-Newton methods are essentially first-order methods, with the underlying idea being that of approximation[74] of the Hessian by gradients. Newton and quasi-Newton methods outperform GD close to x^*, work better with

[71] We assume the functions concerned are always convex and often have additional properties.

[72] Sometimes we will refer to Newton methods to encompass the classical Newton method and its various variants.

[73] I.e., the iterative construction of x_k.

[74] The reason being that for practical purposes the Hessian, being an $n \times n$ matrix, is often too expensive to compute and store.

complicated functions and are generally used on data of smaller size and for complex statistical models. On the other hand, the Newton methods may behave inconveniently for initial conditions far from x^* and typically have expensive iteration steps. When dealing with big amounts of data GD is thus preferable.

Newton methods

Given a point x_k the GD uses $\nabla f(x_k)$ to approximate f by a linear function and extract the direction of the greatest descent. The Newton method on the other hand uses $\nabla f(x_k)$ and $\nabla^2 f(x_k)$ to approximate f by a quadratic function. Since every quadratic has a unique minimum (as long as its Hessian is positive definite), it is naturally to take that minimum as the next iterative step[75]. We start with a motivating example.

Example 6.1. *Define*

$$f(x) = b^T x + \frac{1}{2} x^T H x,$$

where H is a symmetric positive definite matrix[76]*, and compute*

$$\nabla f(x) = Hx + b, \qquad \nabla^2 f(x) = H.$$

The minimum x^* of f satisfies $\nabla f(x^*) = 0$, hence $x^* = -H^{-1}b$.

We now use the insight gained by Example 6.1 to derive the classical Newton method. It is easy to verify[77] that the quadratic function that fits f best at x_k is

$$Q(x_k + t) = f(x_k) + t^T \ \nabla f(x_k) + \frac{1}{2} t^T \ \nabla^2 f(x_k) \ t.$$

Note that $\nabla Q = \nabla f(x_k) + \nabla^2 f(x_k) t$ and so similarly as in Example 6.1, the minimum is attained[78] at

$$t = -(\nabla^2 f(x_k))^{-1} \ \nabla f(x_k),$$

which results in the next best guess x_{k+1} for x^*:

THE NEWTON METHOD:
$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \ \nabla f(x_k)$

The following convergence results shows that the Newton method converges nicely in a neighborhood of x^*.

[75] Although some variants use only the direction towards that minimum and deduce the length of the step via other means, as we will see later.

[76] Without positive semi-definiteness f would have no minimum.

[77] One could use the Taylor expansion to obtain Q or prove by a direct calculation that $Q(x_k) = f(x_k), \nabla Q(x_k) = \nabla f(x_k)$ and $\nabla^2 Q(x_k) = \nabla^2 f(x_k)$.

[78] For this argument we assume $\nabla^2 f(x_k)$ is invertible. Keep in mind that $\nabla^2 f(x_k)$ is a symmetric matrix and $\nabla f(x_k)$ is a vector.

> **Theorem 6.2** (Theorem 3.5 in Nocedal's book)**.** *Suppose f is twice continuously differentiable, $\nabla^2 f$ is L-Lipschitz continuous and positive definite in a neighborhood of x^*. Let x_k be a sequence of iterates constructed via the Newton method. Then there exists $\widetilde{L} > 0$ so that if x_1 is close enough to x^*,*
>
> $$||x_{k+1} - x^*|| \leq \widetilde{L}\ ||x_k - x^*||^2,$$
>
> *i.e., the convergence is quadratic if x_1 is close enough to x^*.*

Proof.

$$x_{k+1} - x^* = x_k - (\nabla^2 f(x_k))^{-1}\ \nabla f(x_k) - x^* \overset{\nabla f(x^*)=0}{=}$$

$$= (\nabla^2 f(x_k))^{-1}\Big(\nabla^2 f(x_k)(x_k - x^*) - (\nabla f(x_k) - \nabla f(x^*))\Big)$$

Using this equality we can express[79]

$$||x_{k+1} - x^*|| =$$

$$= ||(\nabla^2 f(x_k))^{-1} \int_0^1 \Big(\nabla^2 f(x_k) - \nabla^2 f(x^* + t(x_k - x^*))\Big)(x_k - x^*)dt|| \leq$$

$$\leq ||(\nabla^2 f(x_k))^{-1}|| \int_0^1 ||\nabla^2 f(x_k) - \nabla^2 f(x^* + t(x_k - x^*))|| \cdot ||x_k - x^*||dt.$$

As $\nabla^2 f$ is L-Lipschitz[80] we further deduce

$$||x_{k+1} - x^*|| =$$

$$\leq ||(\nabla^2 f(x_k))^{-1}|| \int_0^1 L(1-t)||x_k - x^*|| \cdot ||x_k - x^*||dt =$$

$$= ||(\nabla^2 f(x_k))^{-1}|| \cdot \frac{L}{2} \cdot ||x_k - x^*||^2.$$

We can now choose a radius r so that if $||x - x^*|| < r$ we have $||(\nabla^2 f(x))^{-1}||^{-1} < 2||(\nabla^2 f(x^*))||^{-1}$. Setting $\widetilde{L} = L||(\nabla^2 f(x^*))||^{-1}$ we obtain

$$||x_{k+1} - x^*|| \leq \widetilde{L} \cdot ||x_k - x^*||^2. \tag{4}$$

Choosing x_1 so that $||x_1 - x^*|| \leq \min\{r, \frac{1}{2\widetilde{L}}\}$ we iteratively deduce[81] $||x_k - x^*|| \leq \min\{r, \frac{1}{2\widetilde{L}}\}$ and thus Equation 4 holds for all k. □

While the Newton method often performs well, there are cases where it fails. The most obvious source of problems is the case when the Hessian is not positively definite (Example 6.4), although even the Hessian being positively definite does not guarantee convergence (Example 6.3).

Example 6.3. *Let $f(x) = x^2 - \frac{1}{4}x^4$. By setting $x_1 = \sqrt{2/5}$ the Newton method produces sequence $x_{odd} = \sqrt{2/5}, x_{even} = -\sqrt{2/5}$, hence there is no convergence, even though $f'' > 0$ on $[-\sqrt{2/5}, \sqrt{2/5}]$, see Figure 19.*

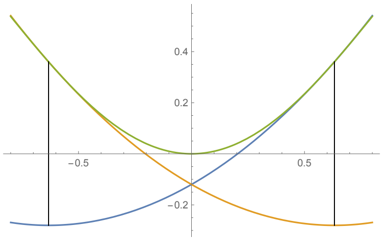

Figure 19: Graph of $f(x) = x^2 - \frac{1}{4}x^4$ in green and two fitted parabolas at $\pm\sqrt{2/5}$ as set in Example 6.3. The Newton method results in a divergent sequence for $x_1 = \sqrt{2/5}$.

Conclusion of Theorem 6.2

$$||x_{k+1} - x^*|| \leq \widetilde{L}\ ||x_k - x^*||^2$$

is referred to as Q-quadratic convergence in Nocedal's book. It implies

$$||x_{k+1} - x^*|| \leq \widetilde{L}^{2^0+2+\ldots+2^{k-1}}\ ||x_1 - x^*||^{2^k},$$

$$||x_{k+1} - x^*|| \leq \widetilde{L}^{2^k-1}\ ||x_1 - x^*||^{2^k}.$$

If $\widetilde{L} > 1$ this means

$$||x_{k+1} - x^*|| \leq (\widetilde{L}\ ||x_1 - x^*||)^{2^k}$$

while for $\widetilde{L} \leq 1$ we get

$$||x_{k+1} - x^*|| \leq\ ||x_1 - x^*||^{2^k}.$$

Both of these conditions imply fast convergence if x_1 is close enough to x^*.

[79] For the first part of the constructed integral note $\int_0^1 \nabla^2 f(x_k)(x_k - x^*)dt = \nabla^2 f(x_k)(x_k - x^*)$. As for the second part,

$$\frac{\partial \nabla f(x^* + t(x_k - x^*))}{\partial t} =$$

$$= \Big(\nabla^2 f(x^* + t(x_k - x^*))\Big)(x_k - x^*)$$

implies

$$\int_0^1 \Big(\nabla^2 f(x^* + t(x_k - x^*))\Big)(x_k - x^*)dt =$$

$$= \nabla f(x^* + t(x_k - x^*))\ |_0^1 =$$

$$= \nabla f(x_k) - \nabla f(x^*).$$

[80] I.e.,

$$||\nabla^2 f(x_k) - \nabla^2 f(x^* + t(x_k - x^*))|| \leq$$

$$\leq L|(x_k - x^* - t(x_k - x^*))| =$$

$$= L(1-t)||x_k - x^*||.$$

[81] Condition $||x_i - x^*|| \leq r$ is required to deduce Equation 4. Together with condition $||x_i - x^*|| \leq \frac{1}{2\widetilde{L}}$ it implies $||x_{i+1} - x^*|| \leq \min\{r, \frac{1}{2\widetilde{L}}\}$.

Example 6.4. *Let $f(x,y) = x^4 + 4xy + (1+y)^2$, see Figure 20. Choosing $x_1 = (0,0)$ we can compute $\nabla f(x_1) = (0,2)^T$ and*

$$\nabla^2 f(x_1) = \begin{pmatrix} 0 & 4 \\ 4 & 2 \end{pmatrix}.$$

By the Newton method we would get $x_2 = (0,0) - (.5,0) = (-.5,0)$ with the step being $-(.5,0)$. Note that this is perpendicular to $\nabla f(x_1)$ hence it is not a descent direction[82] *In fact it can be proved that nowhere along the line of the Newton step does the function decrease, and function value at $(-.5,0)$ is larger than at $(0,0)$.*

While Example 6.4 demonstrates that a single step of a Newton method may result in a less favourable x_2, we could construct a function so that the same would happen in each step and the Newton iteration would not only diverge, but actually increase the value of the function.

We next provide several modifications aimed at overcoming some of the mentioned issues.

Adaptations to the Newton method

1. **The line search**. Instead of using the default step of the Newton method, we could adjust it to any suitable length by adjusting the parameter α:

   $$x_{k+1} = x_k - \alpha(\nabla^2 f(x_k))^{-1}\ \nabla f(x_k), \qquad \alpha \in \mathbb{R}.$$

   This won't solve the issue in Example 6.4, as no choice of α would decrease the function value. However, this approach would adjust the step in Example 6.3 and make it converge. By Lemma 6.5, the direction $-(\nabla^2 f(x))^{-1}\nabla f(x)$ is a descent direction if $\nabla^2 f(x)$ is positive definite, hence some positive $\alpha = \alpha_k$ works in each step and allows us to avoid the pitfall of Example 6.3.

   The main challenge in this case is the choice of α. Typically we would start with $\alpha = 1$, which is the natural choice and the default of the Newton method. We would then verify the Wolfe[83] conditions and if necessary, iteratively decreasing α until the Wolfe conditions are satisfied.

2. **A modification of the Hessian.** The second modification deals with non-positive definite Hessians, an issue which may cause serious issues[84] even if we employ the line search. We present two possible workarounds to change $\nabla^2 f(x_k)$ into a closely related positive definite matrix:

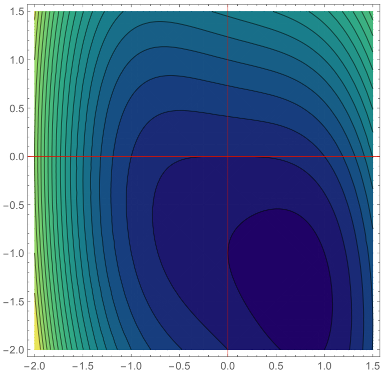

Figure 20: Contour plot of $f(x,y) = x^4 + 4xy + (1+y)^2$ set in Example 6.4. The Newton method starting at $(0,0)$ does not decrease the function value in the first step.

Lemma 6.5. *If $\nabla^2 f(x)$ is positive definite then $-(\nabla^2 f(x))^{-1}\nabla f(x)$ is a descent direction.*

Proof. A direction v at x is a descent direction iff the directional derivative at x along v is negative, i.e., iff $(\nabla f(x))^T v < 0$. In our case this holds as

$$-(\nabla f(x))^T\ (\nabla^2 f(x))^{-1}\ \nabla f(x) < 0.$$

We used the fact that as $\nabla^2 f(x)$ is positive definite, so is its inverse. □

[82] Meaning the corresponding directional derivative is not negative.

[83] A line search is typically performed by verifying the Wolfe (or sometimes Armijo) conditions for a chosen α and then adjusting α if necessary. The Wolfe conditions do not guarantee that we found a minimum along the line of search, but provide a close enough approximation.

[84] As demonstrated by Example 6.4.

(a) Diagonalize $\nabla^2 f(x_k)$, change the sign of all negative eigenvalues and use this modified version of the Hessian in the construction of x_{k+1}. Diagonalization is of course costly and there is in general no guarantee that this solves the issue.

(b) Change $\nabla^2 f(x_k)$ to $\nabla^2 f(x_k) + \lambda I$ for some $\lambda > 0$. While the approach in (a) changes the sign of the negative eigenvalues of $\nabla^2 f(x_k)$, this approach shifts them all by λ. Of course, the smaller λ we choose, the smaller the change will be, which is preferrable. As we want the result to be positive definite, λ should be larger than

$$-\max\{0, \textit{minimal eigenvalue of } \nabla^2 f(x_k)\}.$$

In practice we would typically test whether $\nabla^2 f(x_k) + \lambda I$ is positive definite for some λ and then adjust the parameter if deemed necessary[85]. How do we test whether a matrix is positive definite? We try to perform the Choleski factorization: a matrix is positive definite iff the Choleski factorization algorithm returns a decomposition.

This modification is preferable to (a). There is also a good reason to expect it to work. Notice that the λI term alone represents the gradient descent with step λ. Hence $\nabla^2 f(x_k) + \lambda I$ is a weighted combination of a gradient descent and the Newton method. Since the gradient descent always proceeds along the steepest descent direction, this modification actually adjusts the Newton step towards the steepest descent.

Implementations of a Newton method typically incorporate these two modifications. All of the quasi-Newton methods below are to be used at least with the line search modification.

Quasi-Newton methods

Computing, storing and inverting Hessians[86] requires a lot of resources. Quasi-Newton methods aim to approximate the inverse of a Hessian by a matrix B_k, which typically depends only[87] on gradients. AdaGrad presented in the previous section fits into the framework of quasi-Newton methods.

QUASI-NEWTON METHODS:
$x_{k+1} = x_k - B_k \nabla f(x_k)$

Obviously the difference between various quasi-Newton methods is in construction of B_k. We will present two more methods of constructing B_k. Before we do so we discuss the QN-condition[88] that the

[85] If the resulting matrix is not positive definite we need to increase λ and test again. If the resulting matrix is positive definite we could stick to the current λ or decrease it and test again.

[86] Note that Hessians are typically not sparse.

[87] This means that quasi-Newton methods are actually first order methods trying to imitate second order methods.

[88] I.e., quasi-Newton condition.

quasi-Newton methods typically satisfy.

Recall that the Newton method is based on an approximation of a function f at x_k by a quadratic[89] Q so that:

- $\nabla Q(x_k) = \nabla f(x_k)$ and
- $\nabla^2 Q(x_k) = \nabla^2 f(x_k)$.

Most of the quasi-Newton methods are based on an approximation of a function f at x_k by a quadratic Q so that[90]:

1. $\nabla Q(x_k) = \nabla f(x_k)$ and
2. $\nabla Q(x_{k-1}) = \nabla f(x_{k-1})$.

Let us analize these two conditions. Quadratic Q is of the form[91]

$$Q(x_k + t) = f(x_k) + t^T \nabla f(x_k) + \frac{1}{2} t^T D_k t.$$

Condition 2. above translates[92] to

$$\nabla f(x_{k-1}) = \nabla f(x_k) + D_k(x_{k-1} - x_k).$$

Introducing notation $\gamma_k = \nabla f(x_k) - \nabla f(x_{k-1}), \delta_k = x_k - x_{k-1}$, and $B_k = D_k^{-1}$ we obtain the QN condition:

$$B_k \gamma_k = \delta_k.$$

QN condition is the basis of the following two methods:

1. **SR1: rank 1 update.** The idea of SR1 is to update B_k in each step by a rank 1 matrix while preserving its properties of being positive definite and symmetric. Technically this means[93] there exist $u_k \in \mathbb{R}^n$ so that

   $$B_{k+1} = B_k + u_k u_k^T.$$

   Coupling[94] this equality with the QN condition we obtain

   $$B_k \gamma_{k+1} + u_k u_k^T \gamma_{k+1} = \delta_{k+1}. \tag{5}$$

   As $u_k u_k^T \gamma_{k+1}$ is parallel to u_k we deduce

   $$u_k = p_{k+1}(\delta_{k+1} - B_k \gamma_{k+1})$$

   for some $p_{k+1} \in \mathbb{R}$. Inserting the last equality into Equation (5) results[95] in $p_{k+1}^2 = (\gamma_{k+1}^T(\delta_{k+1} - B_k \gamma_{k+1}))^{-1}$.

$$\boxed{\begin{array}{l} \textit{SR1: For } \delta_{k+1} = \delta, \gamma_{k+1} = \gamma \\ B_{k+1} = B_k + \frac{(\delta - B_k\gamma)(\delta - B_k\gamma)^T}{\gamma^T(\delta - B_k\gamma)} \end{array}}$$

Theorem 6.6. *If f is a strongly convex quadratic with Hessian A, $x_1 \in \mathbb{R}^n$, B_1 is any symmetric positive definite matrix, and $\gamma_{k+1}^T(\delta_{k+1} - B_k\gamma_{k+1}) \neq 0$, then SR1 converges in at most n steps. Furthermore, if the first n search directions are linearly independent, then $B_{n+1} = A^{-1}$.*

[89] An obvious condition is $Q(x_k) = f(x_k)$.

[90] They use only gradients.

[91] Here D_k is a symmetric positive-definite matrix to be defined, that represents an approximation of the Hessian of x_k.

[92] Note that $\nabla Q(x_k + t) = \nabla f(x_k) + D_k t$.

[93] Typically the initial matrix B_1 is a positive multiple of the identity matrix.

[94] I.e., multiplying from right by γ_{k+1}.

[95] Of course, assuming that $\gamma_{k+1}^T(\delta_{k+1} - B_k\gamma_{k+1}) \neq 0$.

Our deduction above implies that this is the only rank 1 update with the stated properties[96]. It turns out to be fairly stable and provides good approximations of the inverse of the Hessian by Theorem 6.6. However, its performance is typically superseded by the rank 2 update below.

2. **BFGS: a rank 2 update.** BFGS is perhaps the most popular quasi-Newton method due to its excellent performance. It is one of the rank 2 update methods satisfying the QN condition. It could be deduced in a similar although more technical way as SR1.

BFGS:

$$B_{k+1} = B_k - \frac{\delta\gamma^T B_k + B_k\gamma\delta^T}{\delta^T\gamma} + \left(1 + \frac{\gamma^T B_k \gamma}{\delta^T\gamma}\right)\frac{\delta\delta^T}{\delta^T\gamma}$$

Both SR1 and BFGS provably converge under favourable curcumstances[97].

Limited memory quasi-Newton methods

Limited memory versions of quasi-Newton methods tackle the problem of storing large typically non-sparse matrices B_k. Here we will present a limited version of the BFGS method called L-BFGS, which sacrifices some performance but significantly reduces the memory usage. Before we go into details let us rephrase BFGS in an inductive matter.

Define $\rho_k = (\delta_k^T\gamma_k)^{-1} \in \mathbb{R}$ and $n \times n$ matrices $V_k = I - \rho_k\gamma_k\delta_k^T$. We can thus express the BFGS inductive step as

$$B_k = V_k^T B_{k-1} V_k + \rho_k \delta_k \delta_k^T.$$

After m iterations[98] we obtain

$$\begin{aligned} B_k = \quad & (V_k^T \dots V_{k-m+1}^T)\ B_{k-m}\ (V_{k-m+1} \dots V_k) + \\ & +\rho_{k-m+1}\ (V_k^T \dots V_{k-m+2}^T)\ \delta_{k-m+1}\delta_{k-m+1}^T\ (V_{k-m+2} \dots V_k) + \\ & \ \vdots \\ & +\rho_{k-1} V_k^T \delta_{k-1}\delta_{k-1}^T V_k + \\ & +\rho_k \delta_k \delta_k^T \end{aligned} \tag{6}$$

L-BFGS incorporates **three modifications** with respect to BFGS:

1. While a matrix of BFGS stores the entire history up to that point, matrix B_k of L-BFGS contains only the history of the last[99] m steps. Matrix B_k can be expressed using Equality (6), substituting the (at this point unsaved) B_{k-m} for an appropriate initialisation[100] matrix $\widetilde{B}_k$. It goes without saying that the first m steps coincide with the BFGS method.

[96] I.e., preserving properties of B_k of being positive definite and symmetric, and satisfying QN conditions.

[97] For details see Section 6.4 in Nocedal's book.

[98] For the sake of simplicity let's assume $k > m$.

[99] Values of m are usually chosen between 3 and 20.

[100] Matrices $\widetilde{B}_k$ should be diagonal, else they undo the performance improvements below. Typically

$$\widetilde{B}_k = \frac{\delta_k^T\gamma_k}{\gamma_k^T\gamma_k} I$$

performs well in practice.

2. While BFGS stores potentially enormous matrices B_k of size $n \times n$, L-BFGS stores these matrices indirectly by storing only the last m pairs δ_k, γ_k. It is apparent from Equality (6) that this information suffices to recreate B_k while reducing the memory consumption to only $2mn$.

3. BFGS performs matrix multiplication $B_k \nabla f(x_k)$, which is potentially costly. L-BFGS instead computes a series of dot products[101] to obtain the same result. Hence matrix B_k is never really computed but rather stored implicitly in δ_k, γ_k, while product $B_k \nabla f(x_k)$ is computed directly from this implicit description.

We consequently obtain the following two-loop recursion algorithm to compute the L-BFGS step direction, which is then incorporated into a master quasi-Newton method along with a line search.

```
Result: $B_k \cdot \nabla f(x_k)$
$q = \nabla f(x_k)$;
for $i = k, \ldots, k - m + 1$ do
    $\alpha_i = \rho_i \delta_i^T q$
    $q = q - \alpha_i \gamma_i$
end
$r = \widetilde{B}_k q$
for $i = k - m + 1, \ldots, k$ do
    $\beta = \rho_i \gamma_i^T r$
    $r = r + \delta_i(\alpha_i - \beta)$
end
return $r$
```

Algorithm 1: L-BFGS step with a two-loop recursion.

[101] Suppose $a, b, c \in \mathbb{R}^n$ and we want to compute ab^Tc. While the final result does not depend on the order of operations, the computational requirements do. Computing as $a(b^Tc)$ we execute one dot product and one vector-scalar product, which requires of the order n operations and storage. Computing as $(ab^T)c$ we eventually multiply a (typically) non-sparse $n \times n$ matrix ab^T by c, which requires of the order n^2 operations and storage.

Summary

We conclude with a summary of presented methods.

1. The Newton method:

   + Performance is great close to the minimum.
   - The method may not converge and may not even provide a descent direction if the Hessian is not positively definite.
   - The method may not converge even if the Hessian is positively definite.
   - Iteration steps are expensive[102].

2. Adaptations to the Newton method:

   + Incorporated line search adjusts the step[103].
   + A modification of the Hessian[104].

3. SR1: rank 1 update:

   + Stable, good performance.
   + Matrices B_k are good approximations of the Hessian.
   +- Faster but less precise than the Newton method.

4. BFGS: rank 2 update:

   + Stable, superior performance.
   + Typically the method of choice.
   +- Faster but less precise than the Newton method.

5. L-BFGS:

   + Fastest, requires much less memory.
   - Discards most of the history.

6 References

[1] Bubeck, S., *Convex optimization: Algorithms and complexity*, Foundations and Trends in Machine Learning 8.3-4 (2015): 231–357.

[2] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[3] Nocedal J. and Wright S., *Numerical Optimization*, Springer-Verlag New York, 2006.

The author would like to thank students **GG, AM, SP, DN, MK, BD**, and **UZ** for providing a series of comments that improved this text.

[102] They require computing the Hessian, storing and inverting it.

[103] Typically improves performance and is standardly used for quasi-Newton methods.

[104] In case of the Hessian not being positive definite this modification usually adjusts the iteration in an appropriate way.