SC 653: Optimisation for Large Scale Machine Learning **Autumn 2023**

Lecture 1 — August 1

Lecturer: Avishek Ghosh *Scribe: Kaishva Chintan Shah*

1.1 Prerequisites

There are no formal prerequisites, but familiarity with the following topics is required.

- Basics of Optimization (Necessary, Sufficient conditions, basics of convex optimizations)
- Probability
- Linear Algebra

Non-trivial results on optimization, linear algebra, probability will be discussed.

1.2 Overall Content

This course is about optimization algorithms

- Machine Learning problems (GD, SGD, ADAM)
  Pytorch uses Adaptive SGD (ADAM) to train NNs.
- Performance of optimization algorithm?
  How do we define performance
- Theoretical analysis for convergence
  Some implementation in HWs (in any language)

The course is roughly divided in 2 parts

Part 1 (Foundational Topics)

- Review of convex optimization, GD, convergence analysis
- Constraint optimization, projected GD, Frank-Wolfe
- Non-smooth optimization, sub-gradients, proximal algorithms
- Stochastic version of GD (SGD), Convergence Rate of SGD for convex/non-convex problems

Part 2 (Advanced/Recent topics)

- Optimization with momentum/accelerated GD
- Adaptive GD (AdaGrad, ADAM)
- Deep Learning: Understanding SGD for convergence of Neural Nets
  Will cover '2-layer NN' convergence

1.3 References

We won't follow any standard textbook. The following are the references (pdfs of all the following are available online):
For Part 1:

- Optimization for Data Analysis; S. Wright and Ben. Recht
  (Rough draft available DOI: https://people.eecs.berkeley.edu/brecht/opt4ml_book//
- Convex Optimization; S Boyd
- Numerical Optimization; Nocedal and Wright

For Part 2:

- Lectures on Convex Optimization; Y. Nesterov
- Research Papers (will be announced before class)

1.3.1 Discrete/continuous optimization

In finite dimensional continuous optimization problems the constraints set is generally given by $C \subseteq \mathbb{R}^d$; for example $C := [0,1]^d$ (Unit Hyperplane). We will be discussing only continuous problems in this course. In discrete optimization problems the constraint sets are discrete in nature; for example $C := \{0,1\}^d$ (Boolean Hyperplane). Here's an example of a continuous optimization problem:

$$\min_{x \in C} \quad f(x) := \frac{1}{2} x^T Q x - b^T x, \tag{1}$$

with the following data:
(a) $x \in \mathbb{R}^d$ is the decision variable, f is the quadratic objective or the cost cost function, and Q is a symmetric positive semi-definite matrix;

(b) the constraint set is given by $C := [0,1]^d$.

1.3.2 Gradient Descent

Reading Exercise:- Directional Derivative and GD

Algorithm 1: Gradient Descent Algorithm

Data : $f(x)$
Initialize: Learning rate α
while *not converged* **do**
  $x^{k+1} = x^k - \alpha^k (Qx^k - b)$
  continue
end
Output : x

Problem: How do we consider the constraint - $[0,1]^d$?
We take the projection of the point onto the provided constraint set; if it ever leaves the set, and continue

with that point

$$x^{k+1} = \mathbb{P}[x^k - \alpha^k(Qx^k - b)](\text{Projected GD}) \tag{2}$$

Last class:

In the last class, introduction to Gradient Descent (GD) algorithm with quadratic loss was discussed. Also, the Projected Gradient Descent (PGD) algorithm was discussed for the case of constraints on the decision variable.

2.4 Deterministic Algorithm

Consider the following quadratic loss function:

$$f(x) := \frac{1}{2}x^T Qx - b^T x \tag{3}$$

where, $x \in \mathbb{R}^d$, $Q \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$. Q is a symmetric and positive semi-definite matrix. As discussed in the previous lecture, the GD algorithm for k^{th} iteration is written as the following,

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k) \tag{4}$$

where $\nabla f(x^k) = Qx^k - b$ and α^k is a parameter. This is an example of a deterministic algorithm.

2.5 Stochastic Algorithm

Suppose, we can only compute the gradient $(\nabla f(x^k))$, then the approximate gradient of the loss function is written as,

$$z^k = \nabla f(x^k) + v^k \tag{5}$$

where, v^k is a random variable with $\mathbb{E}[v^k] = 0$. Then, the Stochastic Gradient Algorithm (SGD) for k^{th} iteration is,

$$x^{k+1} = x^k - \alpha^k z^k. \tag{6}$$

Example 2.5.1. ***Linear Regression***
Supppose, we are given a dataset $(a_i, b_i) \in \mathbb{R}^d \times \mathbb{R}$ for all $i = 1, 2, \ldots n$. Define linear regression cost function,

$$f(x) := \frac{1}{n}\sum_{i=1}^{n} \left(a_i^T x - b_i\right)^2. \tag{7}$$

The goal is to find $x^\star$ such that,

$$x^\star = \arg\min_x f(x). \tag{8}$$

Suppose, we run the GD algorithm to solve the problem (8) *then the gradient of the cost function is,*

$$\nabla f(x^k) = \frac{1}{n}\sum_{i=1}^{n} 2\left(a_i^T x^k - b_i\right) a_i \tag{9}$$

and the update rule is,

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k). \tag{10}$$

The main challenge in this algorithm is, when the number of data points is very high then the number of gradient computation requirement also increases which is computationally expensive.

Algorithm 2: Stochastic Gradient Descent Algorithm

Initialize: x^0
while *error* $>$ *tolerance* **do**
  Select an index ($i^\star$) uniformly at random $i^\star \sim$ Unif $\{1,2,\ldots,n\}$
  $z^k \leftarrow \left[\nabla f(x^k)\right]_{i^\star} = 2\left(a_{i^\star}^T x^k - b_{i^\star}\right) a_{i^\star}$
  $x^{k+1} \leftarrow x^k - \alpha^k z^k$
  $k \leftarrow k+1$
end
Output : $x^\star = x^{k+1}$

Remark 2.1. *Note that the full gradient and average of the gradient computed over all data points are the same, i.e.,*

$$\mathbb{E}[z^k] = \nabla f(x^k). \tag{11}$$

Remark 2.2. *This algorithm is sensitive to the variance of the data. To avoid this, a batched version of the SGD can be implemented.*

2.6 Convex Optimization

Definition 2.3. *A set $C \subseteq \mathbb{R}^d$ is **convex set**, if for all points $x, y \in C$, $\lambda x + (1-\lambda)y \in C$ for any $\lambda \in [0,1]$.*

2.6.1 First Order Optimality Condition for an Unconstrained Convex Optimization

Let a convex function be $f : C \to \mathbb{R}$. Suppose f differentiable at $x^\star$. If $x^\star$ is an optimal solution, then $\nabla f(x^\star) = 0$.

Definition 2.4. *$x^\star$ is a **local minima** for the objective f, if there exists an open set U containing $x^\star$ such that,*

$$f(x^\star) \leqslant f(y) \quad \text{for all } y \in U. \tag{12}$$

Definition 2.5. *$x^\star$ is a **global minima** for the objective f, if*

$$f(x^\star) \leqslant f(y) \quad \text{for all } y \in C. \tag{13}$$

Exercise 2.6. *Prove that, for convex functions, all local minima are global minima.*

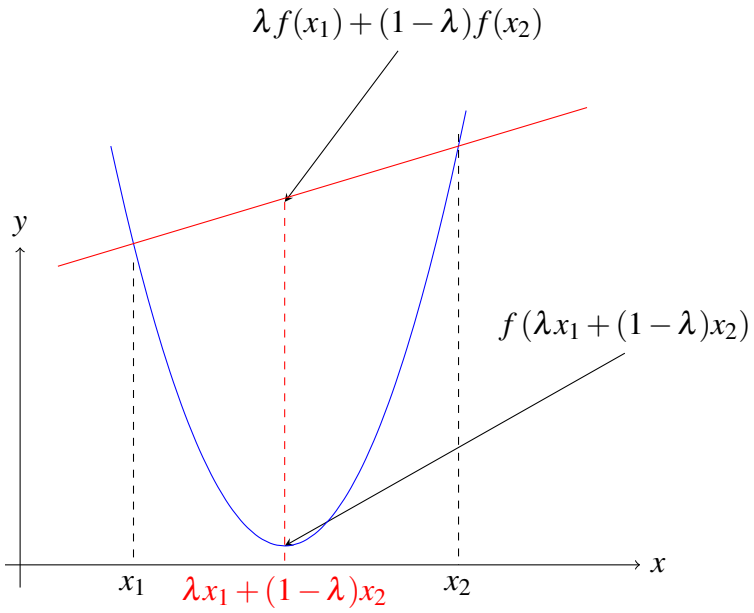


Figure 2.1. A convex function

2.7 Structured Optimization Problem

In this section, we introduce some structure on the objective function and then analyze different algorithms on the problem.

Definition 2.7. *A function $f : X \to Y$ is **Lipchitz continuous function** (with respect to ∞-norm), if there exists a real number L such that for every $x, y \in X$,*

$$|f(x) - f(y)| \leqslant \mathrm{L}\,\|x - y\|_\infty . \tag{14}$$

2.7.1 Optimization using Zeroth-Order Oracle

Definition 2.8. *An n^{th} **order oracle** is outputs n^{th} order information given an input. e.g., zeroth-order oracle outputs $\{f(x)\}$, given an x and first-order oracle outputs $\{f(x), \nabla f(x)\}$, given an x.*

Consider the following optimization problem,

$$\min_{x \in [0,1]^d} f(x) \tag{15}$$

where f is a Lipchitz function.

We intend to solve the problem (15) using zeroth-order oracle. We use a naive approach of discretization. For simplicity, consider the dimension of the decision space as $d = 2$.

2.7.2 Performance Analysis of Zeroth-Order Oracle

Suppose, the true optimizer is $x^\star$ such that, $x^\star = \arg\min_{x \in [0,1]^2} f(x)$. The algorithm 3 outputs $\tilde{x}$. Suppose, $\mathrm{S}_{x^\star}$ is the set of closet/neighbouring grid points of $x^\star$. By construction there exists an $\bar{x} \in \mathrm{S}_{x^\star}$ such that,

$$\|\bar{x} - x^\star\|_\infty \leqslant \frac{\varepsilon}{\mathrm{L}}. \tag{16}$$

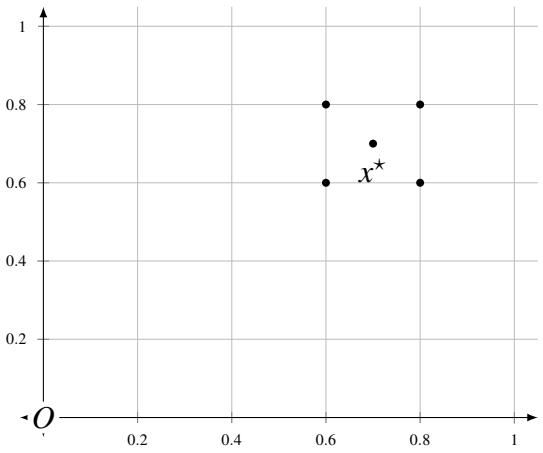


Figure 2.2. Illustration of discretization using grid resolution $\frac{2\varepsilon}{\mathrm{L}}$

Algorithm 3: Discretization Algorithm

Data : $\varepsilon > 0$, L

1 Choose grid resolution = $\frac{2\varepsilon}{\mathrm{L}}$ and let $X :=$ set of all grid points

2 Evaluate $f(x)$ for all $x \in X$

Output: $\tilde{x} = \arg\min_{x \in X} f(x)$

Now, from the Algorithm 3 we have,

$$\begin{aligned} & f(\tilde{x}) \leqslant f(\bar{x}) \\ \implies & f(\tilde{x}) - f(x^\star) \leqslant f(\bar{x}) - f(x^\star) \\ & \leqslant |f(\bar{x}) - f(x^\star)| \\ & \leqslant \mathrm{L}\,\|\bar{x} - x^\star\|_\infty \\ & = \mathrm{L}\frac{\varepsilon}{\mathrm{L}} \\ & = \varepsilon. \end{aligned} \tag{17}$$

Therefore, $f(\tilde{x}) \leqslant f(x^\star) + \varepsilon$.

Remark 2.9. *The number of oracle calls are* $\left(1 + \frac{\mathrm{L}}{2\varepsilon}\right)^d$. *If the dimension* d *is very large, then the algorithm becomes prohibitive with small* ε.

Remark 2.10. *With zeroth-order oracle, no algorithm can solve the problem with less than* $\left(\frac{\mathrm{L}}{2\varepsilon}\right)^{2d}$ *oracle calls.*

2.8 How to characterize the "Goodness" of an algorithm?

Definition 2.11 ([5]). *For a given* $\varepsilon > 0$, *any algorithm* A, *and any objective function* f *define the* ***oracle complexity****,*

$$\mathscr{N}(\varepsilon, A, f) := \min\{\mathtt{K} | f(\tilde{x}) \leqslant f(x^\star) + \varepsilon\} \tag{18}$$

where $\mathtt{K}$ *is the number of oracle calls.*

Definition 2.12. *For a given $\varepsilon > 0$, any algorithm A, and any objective function class $\mathscr{F}$ define the* ***uniform complexity****,*

$$\mathscr{N}(\varepsilon, A, \mathscr{F}) := \sup_{f \in \mathscr{F}} \mathscr{N}(\varepsilon, A, f). \tag{19}$$

Exercise 2.13. *Prove that, for the class of Lipschitz continuous functions ($\mathscr{F}$),*

$$\mathscr{N}(\varepsilon, \text{zeroth-order}, \mathscr{F}) = \left(1 + \frac{\mathrm{L}}{2\varepsilon}\right)^d.$$

Lecture 3 — August 8

Lecturer: Avishek Ghosh *Scribe: Shivam Patel*

Previous Lecture: Oracle Complexity, Lipschitz Continuity
Today's Lecture: Gradient Descent for Quadratic Objective, Convergence Analysis, Generic Smooth Objective

3.9 Gradient Descent for Quadratic Objective

The objective function $f(x)$ for quadratic objective is given as

$$\min_{x \in C} \quad f(x) = \frac{1}{2} x^T Q x - b^T x \tag{20}$$

where $x \in \mathbb{R}^d$ and $\mathbb{Q}$ is a (symmetric) positive semi-definite matrix. This is an example of unconstrained optimization.
We define $\lambda_{max}(Q) = M$ and $\lambda_{min}(Q) = m$ are the maximum and minimum eigenvalues of Q and $M, m > 0$.
We need to find optimal x^* such that

$$\begin{aligned} x^* &= \arg\min_x \frac{1}{2} x^T Q x + b^T x \\ \nabla f(x) &= Qx - b \\ Qx^* &= b \text{ for } \nabla f(x^*) = 0. \end{aligned}$$

Algorithm 4: Gradient Descent Algorithm (Quadratic Loss)

Data : Q, b, α, T
Initialize: Random Vector x^T
for $\ell \in [0, 1, \cdots, T-1]$ **do**
  $x^{\ell+1} = x^\ell - \alpha \nabla f(x^\ell)$
end
Output : Converged vector parameter x^T

3.10 Convergence Analysis

$$\begin{aligned}\nabla f(x^k) = Qx^k - b &= Qx^k - Qx^* \\ &= Q(x^k - x^*)\end{aligned}$$

$$\begin{aligned}x^{k+1} - x^* &= x^k - x^* - \alpha Q(x^k - x^*) \\ &= (\mathbb{I} - \alpha Q)(x^k - x^*)\end{aligned}$$

Taking the ℓ_2 norm

$$||x^{k+1} - x^*||_2 = ||(\mathbb{I} - \alpha Q)(x^k - x^*)||$$

Let the ℓ_2 operator norm $|||R||| = \max||Rx||_2$ subject to $||x||_2 = 1$. This is known as the variational characterization of eigenvalues. $|||R|||_2 = \lambda_{max}(R)$.

$$||x^{k+1} - x^*||_2 \leqslant |||\mathbb{I} - \alpha Q|||\ ||x^k - x^*||$$

To provide convergence guarantees on the gradient descent algorithm, we need to upper bound the expression $|||\mathbb{I} - \alpha Q|||$. The appropriate choice of α for minimizing $|||\mathbb{I} - Q|||$ is $\alpha = \frac{2}{M+m}$.

Lemma 3.14. *The following contraction can be guaranteed for the assumed choice of $\alpha = \frac{2}{M+m}$:*

$$\begin{aligned}|||\mathbb{I} - \alpha Q||| &\leqslant \left[1 - \frac{2}{M+m}\lambda_{min}Q\right] \\ &= \left[1 - \frac{2}{M+m}m\right] \\ &= \frac{M-m}{M+m} = \frac{1 - m/M}{1 + m/M}\end{aligned}$$

With condition number $\kappa = m/M$,

$$\Rightarrow ||x^{k+1} - x^*||_2 \leqslant \left(\frac{1 - 1/\kappa}{1 + 1/\kappa}\right)||x^k - x^*||_2$$

Theorem 3.15. *For quadratic objective $f(\cdot)$, running gradient descent with $\alpha = \frac{2}{M+m}$ implies*

$$||x^k - x^*||_2 \leqslant \left(\frac{1 - 1/\kappa}{1 + 1/\kappa}\right)^k ||x^0 - x^*||_2$$

Proof: The proof is inductive in nature, and can be obtained by unrolling the iterates and applying the contraction factor for each step. □

Remark: We do not know the step size $\alpha = \frac{2}{m+M}$ in practice as M and m are not known. But, they can be empirically estimated by simulations.

3.11 Oracle Complexity

We define the Oracle Complexity $\mathcal{N}(\varepsilon) = \mathcal{N}(\varepsilon, \text{GD}, \text{ Quadratic})$ to the number of iterations after which

$$||x^{\mathcal{N}(\varepsilon)} - x^*|| \leqslant \varepsilon \tag{21}$$

The above definition is more stricter than the general case where

$$\mathcal{N}(\varepsilon) = \min\{k : f(x^k) \leqslant f(x^*) + \varepsilon\}$$

From **Theorem 3.1**,

$$\Big(\frac{1-1/\kappa}{1+1/\kappa}\Big)||x^0 - x^*||_2 \leqslant \varepsilon$$

$$\mathcal{N}(\varepsilon) \leqslant \frac{log\Big(\frac{\varepsilon}{||x^0-x^*||_2}\Big)}{log\Big(\frac{1-1/\kappa}{1+1/\kappa}\Big)} = \frac{log\Big(\frac{||x^0-x^*||_2}{\varepsilon}\Big)}{log\Big(\frac{1+1/\kappa}{1-1/\kappa}\Big)}$$

Consider $\kappa >> 1$ (sufficiently large), then $1 - \frac{1}{\kappa} \approx 1$,

$$\mathcal{N}(\varepsilon) \leqslant \kappa log\Big(\frac{||x^0 - x^*||}{\varepsilon}\Big)$$

Remark: No explicit dependence of $\mathcal{N}(\varepsilon)$ on dimension (unlike zeroth-order optimization). However, an implicit dependence arises through κ as it usually increases with increase in dimension.
Remark: The above choice of α is not optimal. It can be improved by Accelerated methods such as Nesterov Acceleration to obtain a better bound:

$$\mathcal{N}(\varepsilon) \leqslant \sqrt{\kappa} log\Big(\frac{||x^0 - x^*||}{\varepsilon}\Big)$$

3.12 Smooth Functions

We so far looked only at Quadratic objective functions (a subset of smooth functions).
Definition: Suppose $f : \mathbb{R}^d \to \mathbb{R}$ satisfies

$$||\nabla f(x) - \nabla f(y)||_2 \leqslant M||x - y||_2$$

Where $M > 0$, then we call f as M-smooth/ M-Gradient Lipschitz.
Example: Quadratic $f(x) = \frac{1}{2}x^T Qx - b^T x$

$$\begin{aligned} ||\nabla f(x) - \nabla f(y)|| &= ||(Qx - b) - (Qy - b)|| \\ &= ||Q(x - y)|| \leqslant M||x - y|| \end{aligned}$$

Hence Quadratic objective function is M-smooth.

3.12.1 Equivalent Characterization of Smooth Functions

Theorem 3.16. *The following statements are equivalent ($\forall x, y$):*

1. $||\nabla f(x) - \nabla f(y)|| \leqslant M||x-y||$
2. $\langle \nabla f(x) - \nabla f(y), x-y \rangle \leqslant M||x-y||^2$
3. $f(y) - \{f(x) + \langle \nabla f(x), y-x \rangle\} \leqslant \frac{M}{2} \|x-y\|^2$
4. $\langle \nabla f(x) - \nabla f(y), x-y \rangle \geqslant \frac{1}{M}||\nabla f(x) - \nabla f(y)||^2$
5. *f is twice differentiable then:* $-M\mathbb{I} \preccurlyeq \nabla^2 f \preccurlyeq M\mathbb{I}$ *where* $M = \lambda_{max}$ *(i.e. $M\mathbb{I} - \nabla^2 f$ is P.S.D. matrix).*

Proof: For $(1) \Rightarrow (2)$, we observe that inner product is connected to norm by Cauchy Schwarz inequality.

$$\begin{aligned} \langle \nabla f(x) - \nabla f(y), x-y \rangle &\leqslant ||\nabla f(x) - \nabla f(y)||\ ||x-y|| \\ &\leqslant M\ ||x-y||^2 \end{aligned}$$

□

For $(2) \Rightarrow (3)$, we use the Interpolation method where $G:\ [0,1] \to \mathbb{R}$,

$$G(t) = f(x + t(y-x)) - f(x) - \langle \nabla f(x), t(y-x) \rangle$$

Observe that $G(1) =$ LHS $of(3)$ and $G(0) = 0$.

By the fundamental law of calculus,

$$G(1) - G(0) = \int_0^1 G'(t)dt$$

$$\begin{aligned} \text{LHS of } (3) &= \int_0^1 \langle \nabla f(x + t(y-x)), (y-x) \rangle - 0 - \langle \nabla f(x), y-x \rangle\ dt \\ &= \int_0^1 \langle \nabla f(x + t(y-x)) - \nabla f(x), y-x \rangle\ dt \\ &= \int_0^1 \langle \nabla f(x + t(y-x)) - \nabla f(x), t(y-x) \rangle \frac{dt}{t} \end{aligned}$$

Using (2),

$$\begin{aligned} G(1) - G(0) &\leqslant \int_0^1 Mt||x-y||^2 dt \\ &= M||x-y||^2 \int_0^1 t\ dt = \frac{M}{2}||x-y||^2 \end{aligned}$$

For $(3) \Rightarrow (5)$, we utilize the Taylor Series Expansion.
For $(4) \Rightarrow (1)$, we utilize the Cauchy Schwartz Inequality.

Last class:

In the last class, we discussed the gradient descent algorithm for quadratic functions and also discussed the characterization of the smooth functions.

4.13 Strong Convexity

From the last class, $f: C \subset \mathbb{R}^d \longrightarrow \mathbb{R}$, is smooth if one of the following is true.

1. For some fixed $M > 0$, $\|\nabla f(x) - \nabla f(y)\| \leqslant M \|x-y\|$, for all $x, y \in C$.
2. $f(y) - \big(f(x) + \langle \nabla f(x), y-x \rangle\big) \leqslant \frac{M}{2} \|x-y\|^2$.
3. $\nabla^2 f(x) \leqslant M \mathrm{I}_d$.

Definition 4.17 (Strong Convexity). *Let $C \subset \mathbb{R}^d$ be a nonempty convex set then $f: C \longrightarrow \mathbb{R}$ is an m-strongly convex function if there exists $m > 0$ such that for all $x, y \in C$ and $\lambda \in [0,1]$*

$$f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda) f(y) - \frac{m}{2} \lambda (1-\lambda) \|x-y\|^2$$

Remark 4.18. *m-Strong convexity is a stronger notion than convexity. While convexity implies that the function value between x and y is equal or less than the line joining $f(x)$ and $f(y)$, m-strong convexity quantifies this notion and lower bounds the difference between the function value and line joining the points (see Figure 4.3).*

Theorem 4.19. *Let $C \subset \mathbb{R}^d$ be a nonempty convex set and $f: C \longrightarrow \mathbb{R}^d$ is continuously differentiable, then the following are equivalent*

1. *f is m-strongly convex.*
2. *$\|\nabla f(x) - \nabla f(y)\| \geqslant m \|x-y\|$ for all $x, y \in C$.*
3. *$\langle \nabla f(x) - \nabla f(y), x-y \rangle \geqslant m \|x-y\|^2$ for all $x, y \in C$.*
4. *$f(y) - \big(f(x) + \langle \nabla f(x), y-x \rangle\big) \geqslant \frac{m}{2} \|x-y\|^2$, for all $x, y \in C$.*
5. *If f is continuously differentiable $\langle \nabla f(x) - \nabla f(y), x-y \rangle \leqslant \frac{1}{m} \|\nabla f(x) - \nabla f(y)\|^2$, for all $x, y \in C$.*
6. *If f is twice differentiable then $\nabla^2 f(x) \geqslant m \mathrm{I}_d$ for all $x, y \in C$.*

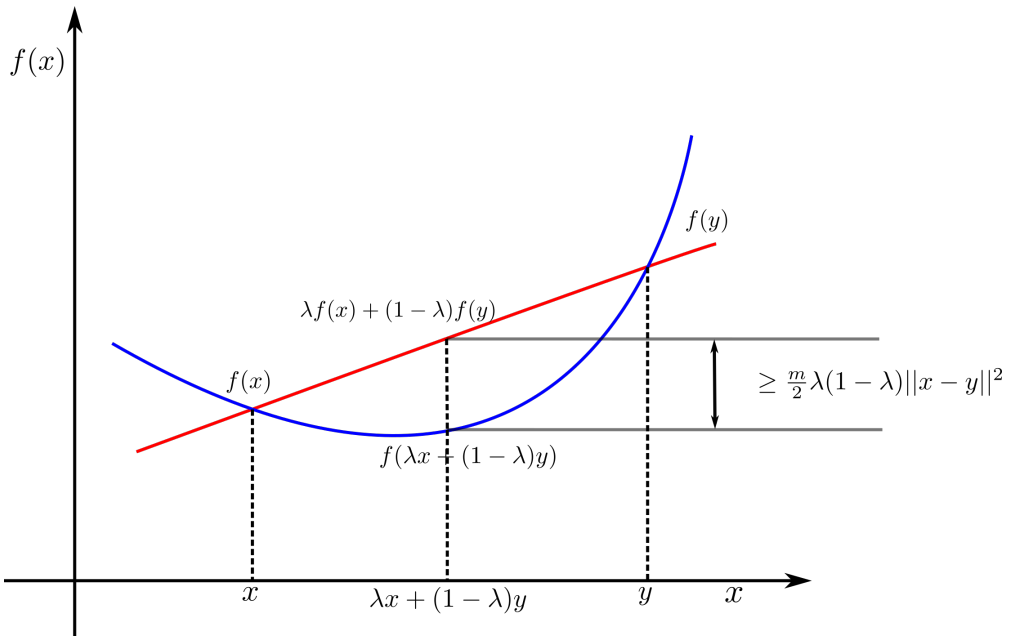


Figure 4.3. m-strongly convex function f

Proof: Define $g(x) = f(x) - \frac{m}{2}\left\|x^2\right\|$ then f is m-strongly convex if and only if g is convex. Then from conditions of convexity, we have

$$\begin{aligned} 0 &\geqslant \|\nabla g(x) - \nabla g(y)\| \quad \text{(from convexity of } g\text{)} \\ &= \|\nabla f(x) - mx - \nabla f(y) + my\| \\ &\geqslant \|\nabla f(x) - \nabla f(y)\| - m\|x-y\| \quad \text{(using triangle inequality)} \\ \|\nabla f(x) - \nabla f(y)\| &\geqslant m\|x-y\| \end{aligned}$$

and thus 1 $\implies$ 2. 2 can be rewritten as

$$\|x-y\| \leqslant \frac{1}{m}\|\nabla f(x) - f(y)\| \quad \text{for all } x, y \in C$$

Further from CBS inequality we have

$$\begin{aligned} \langle \nabla f(x) - \nabla f(y), x-y\rangle &\leqslant \|\nabla f(x) - \nabla f(y)\|\,\|x-y\| \\ &\leqslant \frac{1}{m}\|\nabla f(x) - \nabla f(y)\|^2 \end{aligned}$$

thus 2 $\implies$ 5.
Similarly, we have

$$g(y) - (g(x) + \langle \nabla g(x), y-x\rangle) \leqslant 0$$

Substituting $g(x) = f(x) - \frac{m}{2}\|x\|^2$ gives us 4 and therefore 1 $\implies$ 4. Further from 4, for all $x, y \in C$ we have

$$f(y) - \left(f(x) + \langle \nabla f(x), y-x\rangle\right) \geqslant \frac{m}{2}\|x-y\|^2 \tag{22}$$

and

$$f(x) - \left(f(y) + \langle \nabla f(y), x-y\rangle\right) \geqslant \frac{m}{2}\|x-y\|^2 \tag{23}$$

Adding (37) and (23) we have

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geqslant m \|x - y\|^2$$

and thus $4 \implies 3$. From properties of convexity, if f is continuously differentiable up to the second order then so is g, and thus we have

$$\nabla^2 g(x) = \nabla^2 f(x) - m\mathrm{I}_d$$

is positive definite which implies $\nabla f(x) \geqslant m\mathrm{I}_d$ and thus $1 \implies 6$. Form 6, $\nabla^2 g(x) = \nabla^2 f(x) - m\mathrm{I}_d$ is and thus is positive definite, therefore g is convex and thus f is m-strongly convex thus $6 \implies 1$ □

4.14 Examples

Here are some examples (and non-examples) of strongly convex functions.

Example 4.14.1. *Consider $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, $f(x) = \frac{1}{2}x^\top Qx - b^\top x$, where $Q \in \mathbb{R}^{d\times d}$ is a symmetric positive definite (PD) matrix and $b \in \mathbb{R}^d$. Then we have*

$$\begin{aligned}\|\nabla f(x) - \nabla f(y)\| &\leqslant \|Q(x-y)\| \\ &\leqslant \lambda_{\min}(Q)\,\|x-y\|\end{aligned}$$

where, $\lambda_{\min}$ is the smallest eigen value of Q. Setting $m = \lambda_{\min} > 0$ (since Q is PD) from Theorem 4.19, f is $\lambda_{\min}$-strongly convex.

Example 4.14.2. *Consider $f : [0,\infty) \to \mathbb{R}$, $f(x) = log(1+e^x)$. For such a function we have*

$$\nabla f(x) = \frac{e^x}{1+e^x}$$

and

$$\nabla^2 f(x) = \frac{e^x}{(1+e^x)^2} \geqslant 0$$

Thus f is convex, however for no $\varepsilon > 0$ we have $\nabla^2 f(x) > \varepsilon$ for all $x \in [0,\infty)$ and is therefore not strongly convex. However restricting $x \to [0,\beta]$ for any $0 < \beta < \infty$ we can find an m such that $\nabla^2 f(x) < m$ for all $x \in [0,\beta]$ and thus is strongly convex. Further since $\nabla^2 f(x) \leqslant 1/4$ for all $x \in [0,\infty)$, f is M-smooth, with $M = 1/4$.

Example 4.14.3. *Consider $f\colon [-1,1] \longrightarrow \mathbb{R}$, $x \mapsto f(x) = x^4$, then we have*

$$0 \leqslant \nabla^2 f(x) := 12x^2 \leqslant 12$$

and thus is convex and smooth. However since $\nabla^2 f(0) = 0$ thus it is not strongly convex.

Example 4.14.4 (Invex Functions). *$f\colon [-1/2,\infty) \times [1,\infty) \longrightarrow \mathbb{R}$, $(x,y) \mapsto f(x,y) := y(x^2 - 1^2)$. Computing the hessian for f we have*

$$\nabla^2 f(x,y) = \begin{bmatrix} 4y(3x^2-1) & 4x(x^2-1) \\ 4x(x^2-1) & 0 \end{bmatrix}$$

which is not positive definite for all $(x,y) \in [-1/2,\infty) \times [1,\infty]$ thus it is not convex. However one can show that for any bounded subset of $[-1/2,\infty) \times [1,\infty)$, f is smooth (as f is twice differentiable).

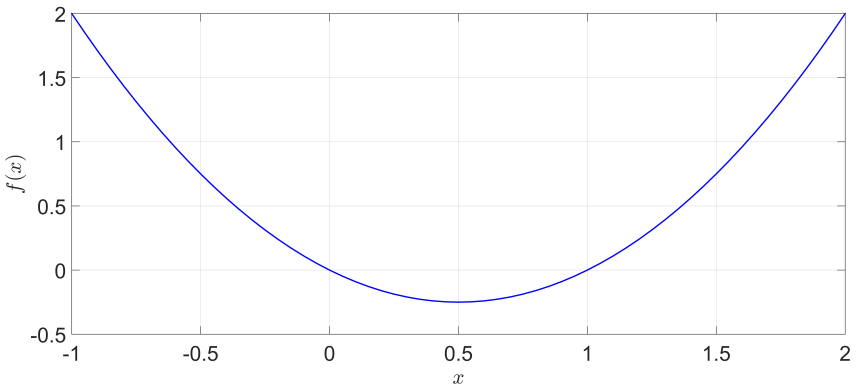


Figure 4.4. $f(x) = \frac{1}{2}xQ^\top x - b^\top x$, with $x \in \mathbb{R}$ and $Q = 2$, $b = 1$: Strongly Convex and smooth

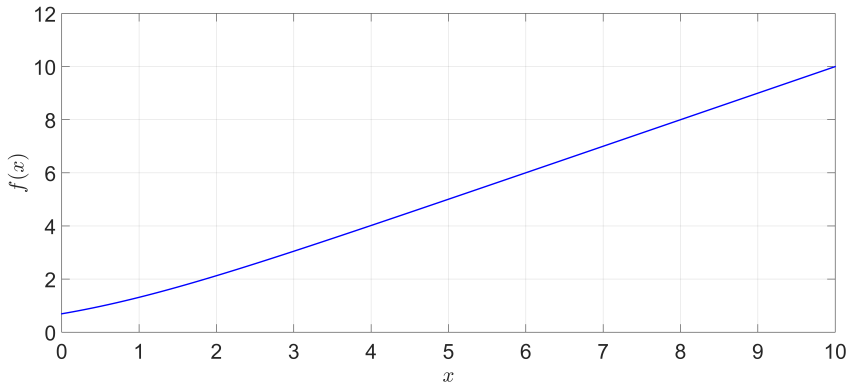


Figure 4.5. $f(x) = log(1+e^x)$, with $x \in [0,\infty)$: Convex and smooth, not strongly convex

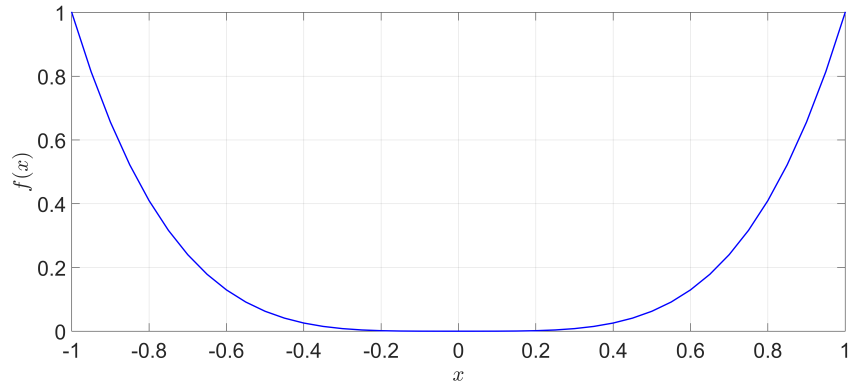


Figure 4.6. $f(x) = x^4$, with $x \in [-1,1]$: Convex and smooth, not strongly convex

4.15 Gradient Descent for Smooth Functions

We are looking at the following optimization problem

$$\min_{x\in\mathbb{R}^d} f(x) \tag{24}$$

where $f:\mathbb{R}^d \longrightarrow \mathbb{R}$ is smooth. Under the assumption that f has a global lower bound we have the following result for gradient descent algorithms for smooth functions

Lemma 4.20 (Descent Lemma). *Let $f:\mathbb{R}^d \longrightarrow \mathbb{R}$ is M-smooth. Implementing gradient descent algorithm*

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

with $\alpha^k = 1/M$, we have

$$f(x^{k+1}) \leqslant f(x^k) - \frac{1}{2M}\|\nabla f(x)\|^2 \leqslant f(x^k).$$

Proof: Since f is M-smooth, for all $x,y\in\mathbb{R}^d$

$$f(y) \leqslant f(x) + \langle \nabla f(x), y-x\rangle + \frac{M}{2}\|x-y\|^2.$$

Setting $x := x^k$ and $y := x^{k=1}$ we have

$$\begin{aligned} f(x^{k+1}) &\leqslant f(x^k) - \left\langle \nabla f(x^k), \alpha^k \nabla f(x^k)\right\rangle + \frac{M}{2}\left\|\alpha^k \nabla f(x^k)\right\|^2 \\ &= f(x^k) - \frac{1}{2M}\left\|\nabla f(x^k)\right\|^2 \\ &\leqslant f(x^k) \end{aligned}$$

thereby completing the proof. □

Theorem 4.21. *Let $f: C \longrightarrow \mathbb{R}^d$ be an M-smooth function, then for gradient descent algorithm after T iterations we have*

$$\min_{0\leqslant k\leqslant T-1}\left\|\nabla f(x^k)\right\| \leqslant \sqrt{\frac{2mf(x^0)-f^*}{T}}$$

where x^0 is the initial guess and f^* is the global lower bound of f on C. Furthermore,

$$\lim_{T\longrightarrow\infty}\left\|\nabla f(x^T)\right\| = 0.$$

Lecture 5 — August 18

Lecturer: Avishek Ghosh *Scribe: Vatsal Kedia*

5.16 Review of last class:

- Smooth functions
- Strongly convex functions
- Descent lemma

In this lecture we are interested in the following unconstrained optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \tag{25}$$

We will look at three different cases by adding structure to the function f above, namely:

1. Smooth function
2. Smooth and Convex
3. Smooth and Strongly convex

First we review the Descent lemma [9].

Lemma 5.22. *If function f is M-smooth and we run GD with $\alpha_k = \frac{1}{M}$ then,*

$$f(x^{k+1}) \leqslant f(x^k) - \frac{1}{2M}\left\|\nabla f(x^k)\right\|^2 \tag{26}$$

5.17 Case 1: Function f is M-smooth but non-convex

Theorem 5.23. *After T-iterations of GD with $\alpha_k = \frac{1}{M}$,*

$$\min_{0\leqslant k\leqslant T-1}\left\|\nabla f(x^k)\right\| \leqslant \sqrt{\frac{2M(f(x^0)-\bar{f})}{T}} \tag{27}$$

Furthermore, we also have the following assertion:

$$\lim_{T\to\infty}\left\|\nabla f(x^T)\right\| = 0 \tag{28}$$

Remark 5.24. *Several remarks are in order:*

1. *The above result is a first-order necessary condition (FONC) guarantee.*
2. *In general for non-convex functions, this is the best one can establish for GD.*
3. *In the above theorem, we don't have any condition on the final iterate but* (28).

Proof: Use Lemma 5.22 with $k = T-1$,

$$f(x^T) \leqslant f(x^{T-1}) - \frac{1}{2M}\left\|\nabla f(x^{T-1})\right\|^2$$

using Lemma 5.22 again with $k = T-1$ on the first term of RHS,

$$f(x^T) \leqslant f(x^{T-2}) - \frac{1}{2M}\left\|\nabla f(x^{T-2})\right\|^2 - \frac{1}{2M}\left\|\nabla f(x^{T-1})\right\|^2$$

It is easy to see that by using Lemma 5.22 iteratively we get,

$$f(x^T) \leqslant f(x^0) - \frac{1}{2M}\sum_{k=0}^{T-1}\left\|\nabla f(x^k)\right\|^2$$

Using assumption, $\exists\ \bar{f}$ such that $\bar{f} \leqslant f(x^k)\ \forall k$ implies $\bar{f} \leqslant f(x^T)$. Therefore,

$$\begin{aligned}\bar{f} &\leqslant f(x^T)\\ &\leqslant f(x^0) - \frac{1}{2M}\sum_{k=0}^{T-1}\left\|\nabla f(x^k)\right\|^2\\ \Rightarrow \sum_{k=0}^{T-1}\left\|\nabla f(x^k)\right\|^2 &\leqslant 2M(f(x^0)-\bar{f})\end{aligned}$$

Case (a): Asymptotic analysis i.e. $T \to \infty$

$$\lim_{T\to\infty}\sum_{k=0}^{T-1}\underbrace{\left\|\nabla f(x^k)\right\|^2}_{\geqslant 0}$$

Sum of infinite non-negative quantity is upper bounded by some finite quantity.

$$\Rightarrow \lim_{T\to\infty}\left\|\nabla f(x^T)\right\|^2 = 0$$

Case (b): Non-asymptotic analysis i.e. $T < \infty$

$$\begin{aligned}\sum_{k=0}^{T-1}\left\|\nabla f(x^k)\right\|^2 &\leqslant 2M(f(x^0)-\bar{f})\\ \Rightarrow \frac{1}{T}\sum_{k=0}^{T-1}\left\|\nabla f(x^k)\right\|^2 &\leqslant \frac{2M}{T}(f(x^0)-\bar{f})\end{aligned}$$

Using property that minimum of non-negative sum term is upper-bounded by average sum.

$$\begin{aligned}\min_{0\leqslant k\leqslant T-1}\left\|\nabla f(x^k)\right\|^2 &\leqslant \frac{1}{T}\sum_{k=0}^{T-1}\left\|\nabla f(x^k)\right\|^2\\ &\leqslant \frac{2M}{T}(f(x^0)-\bar{f})\end{aligned}$$

□

Remark 5.25. *1. Convergence rate is* $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$

2. Initially take large T *and then use early stopping.*

5.18 Case 2: Function f is M-smooth and convex

Theorem 5.26. *Suppose x^* is a minima of f. We run GD with $\alpha_k = \frac{1}{M}$ then,*

$$f(x^T) - f(x^*) \leqslant \frac{M}{2T} \left\|x^0 - x^*\right\|^2 \tag{29}$$

Remarks:

1. Convergence-rate is $\mathscr{O}\left(\frac{1}{T}\right)$.
2. Ensures global convergence.
3. Guarantees on the final iterate and not on the gradient.
4. We assume that f is differentiable since we are using GD otherwise we need sub-gradient methods (will be covered later in the course).

Proof: Given f is convex, then $\forall\, x, y$

$$f(y) \geqslant f(x) + \langle \nabla f(x), y - x \rangle$$

In particular take, $y = x^*$ and $x = x^k$ then:

$$\begin{aligned} f(x^*) &\geqslant f(x^k) + \left\langle \nabla f(x^k), x^* - x^k \right\rangle \\ \Rightarrow f(x^k) &\leqslant f(x^*) - \left\langle \nabla f(x^k), x^* - x^k \right\rangle \end{aligned}$$

Using Lemma 5.22,

$$\begin{aligned} f(x^k) &\leqslant f(x^*) - \left\langle \nabla f(x^k), x^* - x^k \right\rangle - \frac{1}{2M} \left\|\nabla f(x^{T-1})\right\|^2 \\ &= f(x^*) + \left\langle \nabla f(x^k, x^k - x^* \right\rangle - \frac{1}{2M} \left\|\nabla f(x^{T-1})\right\|^2 \\ &= f(x^*) + \frac{M}{2}\left(\left\|x^k - x^*\right\|^2 - \left\|x^k - x^* - \frac{1}{M}\nabla f(x^k)\right\|^2 \right) \\ &= f(x^*) + \frac{M}{2}\left(\left\|x^k - x^*\right\|^2 - \left\|x^{k+1} - x^*\right\|^2 \right) \end{aligned}$$

By summing over $k = 0, 1, \ldots, T-1$, we get

$$\begin{aligned} \sum_{k=0}^{T-1} \left(f(x^k) - f(x^*)\right) &\leqslant \frac{M}{2} \sum_{k=0}^{T-1} \left(\left\|x^k - x^*\right\|^2 - \left\|x^{k+1} - x^*\right\|^2 \right) \\ &= \frac{M}{2}\left(\left\|x^0 - x^*\right\|^2 - \underbrace{\left\|x^T - x^*\right\|^2}_{\geqslant 0} \right) \\ &\leqslant \frac{M}{2} \left\|x^0 - x^*\right\|^2 \end{aligned}$$

$$\Rightarrow \frac{1}{T} \sum_{k=0}^{T-1} \left(f(x^k) - f(x^*)\right) \leqslant \frac{M}{2T} \left\|x^0 - x^*\right\|^2$$

Now, $\{f(x^k)\}_k$ is a non-increasing sequence due to the Descent lemma, hence any term will be upper bounded by average quantity i.e.

$$\begin{aligned} f(x^T) - f(x^*) &\leqslant \frac{1}{T}\sum_{k=0}^{T-1}\big(f(x^k) - f(x^*)\big) \\ &\leqslant \frac{M}{2T}\left\|x^0 - x^*\right\|^2 \end{aligned}$$

□

5.19 Case 3: Function f is M-smooth and m- strongly convex

Lemma 5.27. *If f is m-strongly convex and let x^* is a minima of f then*

$$\|\nabla f(x)\|^2 \leqslant 2m\big(f(x) - f(x^*)\big) \tag{30}$$

Proof: Given f is m-strongly convex, then $\forall\, x, y$

$$f(y) \geqslant f(x) + \langle \nabla f(x), y - x\rangle + \frac{m}{2}\|y - x\|^2.$$

Fix x and optimize over y then

$$\begin{aligned} \nabla f(x) + \frac{m}{2}\big(2(y-x)\big) &= 0 \\ \Rightarrow y = x - \frac{1}{m}\nabla f(x). \end{aligned}$$

Substitute the value of y in RHS of above equation and LHS in $f(x^*)$,

$$\begin{aligned} f(x^*) &\geqslant f(x) + \left\langle \nabla f(x), -\frac{1}{m}\nabla f(x)\right\rangle + \frac{m}{2}\left\|-\frac{1}{m}\nabla f(x)\right\|^2 \\ &= f(x) - \frac{1}{m}\langle \nabla f(x), \nabla f(x)\rangle + \frac{1}{2m}\|\nabla f(x)\|^2 \\ &= f(x) - \frac{1}{2m}\|\nabla f(x)\|^2 \\ \Rightarrow \|\nabla f(x)\|^2 &\geqslant 2m\big(f(x) - f(x^*)\big). \end{aligned}$$

□

Theorem 5.28. *Let f be M-smooth and m-strongly convex. We run GD with $\alpha_k = \frac{1}{M}$ then,*

$$f(x^T) - f(x^*) \leqslant \left(1 - \frac{m}{M}\right)^T \big(f(x^0) - f(x^*)\big). \tag{31}$$

Proof: Using Lemma 5.22,

$$
\begin{aligned}
f(x^{k+1}) &\leqslant f(x^k) - \frac{1}{2M}\left\|\nabla f(x^k)\right\|^2 \\
&\text{applying Lemma 5.27} \\
&\leqslant f(x^k) - \frac{m}{M}\big(f(x^k) - f(x^*)\big) \\
\Rightarrow f(x^{k+1}) - f(x^*) &\leqslant f(x^k) - \frac{m}{M}\big(f(x^k) - f(x^*)\big) - f(x^*) \\
&= \left(1 - \frac{m}{M}\right)\big[f(x^k) - f(x^*)\big] \\
&\leqslant \left(1 - \frac{m}{M}\right)^2\big[f(x^{k-1}) - f(x^*)\big] \\
&\vdots \\
&\leqslant \left(1 - \frac{m}{M}\right)^k\big[f(x^0) - f(x^*)\big]
\end{aligned}
$$

□

6.20 Constrained Optimization

Consider the following constrained optimization problem:

$$\min_{x\in\Omega} \quad f(x) \tag{32}$$

where f is M-smooth, Ω is a closed convex set.

Definition 6.29. *Let Ω be a closed convex set. At any $x \in \Omega$, we define*

$$N_\Omega = \{z \in \mathbb{R}^{\mathrm{d}} \,;\, \langle z, y-x\rangle \leqslant 0 \,, \forall y \in \Omega\} \tag{33}$$

Theorem 6.30. *If $x^\star$ is a local minima, then*

$$-\nabla f(x^\star) \in N_\Omega(x^\star) \tag{34}$$

Furthermore, if f is convex then $x^\star$ is a global minima.

The above theorem is known as the First Order Necessary Condition (FONC) for the optimization problem given in eq(32).

Remark 6.31. *If $\Omega = \mathbb{R}^d$, then $N_\Omega(x^\star) = \{0\}$*

Proof: Suppose $x^\star$ is a local minima. Take $z \in \Omega$. Now since Ω is convex

$$\lambda z + (1-\lambda)x^\star \in \Omega \;\text{ for } \lambda \in [0,1], x^\star + \lambda(z-x^\star) \;\in \Omega. \tag{35}$$

Using the Taylor's series expansion one can write

$$f\big(x^\star + \lambda(z-x^\star)\big) = f(x^\star) + \langle \nabla f(x^\star), \lambda(z-x^\star)\rangle + o(\lambda) \tag{36}$$

where $o(\lambda)$is defined as $\lim_{\lambda\to 0} \frac{o(\lambda)}{\lambda} = 0$. If $x^\star$ is a minima then $f(x^\star + \lambda(z-x^\star)) \geqslant f(x^\star)$. Hence

$$\begin{aligned}
&\langle \nabla f(x^\star), \lambda(z-x^\star)\rangle + o(\lambda) \geqslant 0 \\
\implies &\langle \nabla f(x^\star), (z-x^\star)\rangle + \frac{o(\lambda)}{\lambda} \geqslant 0 \\
\implies &\lim_{\lambda\to 0} \langle \nabla f(x^\star), (z-x^\star)\rangle \geqslant 0 \\
\implies &\lim_{\lambda\to 0} \langle -\nabla f(x^\star), (z-x^\star)\rangle \leqslant 0 \\
\implies &-\nabla f(x^\star) \in N_\Omega(x^\star)
\end{aligned} \tag{37}$$

□

Now, if f is convex for any $z \in \Omega$ and it follows from eq.((37))

$$\begin{aligned}
&f(z) \geqslant f(x^\star) + \langle \nabla f(x^\star), (z-x^\star)\rangle \,, \\
\implies &f(z) \geqslant f(x^\star).
\end{aligned}$$

6.21 Euclidean Projection

Consider Ω to be a closed and convex set. To project x on Ω, we define the projection map as

$$\mathbb{P}_\Omega(x) = \frac{1}{2}\arg\min_{z\in\Omega} \|z-x\|^2 . \tag{38}$$

6.21.1 Properties of $\mathbb{P}_\Omega(.)$

From FONC,

$$-\left(\mathbb{P}_\Omega(x)-x\right) \in N_\Omega\left(\mathbb{P}_\Omega(x)\right). \tag{39}$$

6.21.2 Examples

- **Non-negative Orthant**
  Consider the set Ω as follows

  $$\Omega := \{x\,;\, x_i \geqslant 0,\ \forall\, i = 1,\cdots,d\} \tag{40}$$

  The projection on Ω can be defined as

  $$\mathbb{P}_\Omega(y)\big|_i = \max\{0, y_i\} \tag{41}$$

- **Box**

  $$\Omega := \{x \in \mathbb{R}^d\,;\, \|x\|_\infty \leqslant 1\} \tag{42}$$

  Find the projection on Ω (Exercise).

- L_2 **norm ball**
  Consider the set Ω as

  $$\Omega := \{x \in \mathbb{R}^d\,;\, \|x\|_2 \leqslant 1\} \tag{43}$$

  where the projection on Ω can be defined as

  $$\mathbb{P}_\Omega(y) = \begin{cases} y\,, & \text{for } \|x\|_2 \leqslant 1 \\ \frac{y}{\|y\|_2}\,, & \text{otherwise} \end{cases} \tag{44}$$

- L_1 **norm ball**
  Consider the set Ω as

  $$\Omega := \{x \in \mathbb{R}^d\,;\, \|x\|_1 \leqslant R\} \tag{45}$$

  Show that $\mathbb{P}_\Omega(y)\big|_j = \max\{|y_j| - \lambda, 0\}.\text{sign}(y_j)$ where λ is chosen such that $\|\mathbb{P}_\Omega(y)\|_1 \leqslant 1$.

- **Matrix-Problems**
  Consider the set Ω defined as

  $$\Omega := \{x \in \mathbb{R}^{d\times d}\,;\, x = x^T, x \geqslant 0\} \tag{46}$$

  where distance between two points $x, Y \in \Omega$ can be defined by Frobenius norm as

  $$\|x-Y\|_F^2 = \sum_{i,j}\left(x_{ij} - Y_{ij}\right)^2 \tag{47}$$

6.22 Projected Gradient Descent (PGD)

Algorithm:
At k^{th} iterate,

$$x^{k+1} = \mathbb{P}_{\Omega}\left[x^k - \alpha^k \nabla f(x^k)\right]. \tag{48}$$

Lemma 6.32. *Consider f is M-smooth, Ω is closed and convex. Then*

$$-\nabla f(x^\star) \in N_{\Omega}(x^\star) \text{ iff } x^\star = \mathbb{P}_{\Omega}\left(x^\star - \alpha \nabla f(x^\star)\right)$$

.

Proof: For the forward direction consider that $x^\star$ satisfies FONC. Then we have for $\alpha > 0$ and for all $z \in \Omega$

$$-\alpha \left\langle \nabla f(x^\star), z - x^\star \right\rangle = \left\langle x^\star - \alpha \nabla f(x^\star) - x^\star, z - x^\star \right\rangle. \tag{49}$$

Hence, we get

$$\begin{aligned} &\langle \mathbb{P}_{\Omega}(x) - x, z - x \rangle \geqslant 0 \\ \implies &x^\star = \mathbb{P}_{\Omega}\left(x^\star - \alpha \nabla f(x^\star)\right). \end{aligned} \tag{50}$$

□

Lecture 7 — August 25

Lecturer: Avishek Ghosh *Scribe: Siddhant Vibhute*

Announcements:

- HW 1 will be released on 01 September, Friday
- Deadline for submission : 12 September, Tuesday in class

Review of last class:

- First order necessary condition for constrained optimization
- Convex analysis for constrained problems

Agenda for today's class:

- Projected Gradient Descent (PGD)
- Guarantees on the PGD algorithm

7.23 Constrained optimization

We are interested in the following problem

$$\min_{x\in\Omega} f(x), \tag{51}$$

where $\Omega \in \mathbf{R}^n$ is closed and convex and f is smooth.

Euclidean projection:
The Euclidean projection of a point x onto Ω is the closest point in Ω to x.

$$P_\Omega(x) = \arg\min_{y\in\Omega} \frac{1}{2}\|y-x\|^2 \tag{52}$$

We argued that $P_\Omega(x)$ is unique and well-defined.

First order necessary condition on $P_\Omega(x)$:

$$x - P_\Omega(x) \in N_\Omega\left(P_\Omega(x)\right) \tag{53}$$

where $N_\Omega(P_\Omega(x))$ is the normal cone at $P_\Omega(x)$.
Invoking the definition of the normal cone, we get

$$\langle x - P_\Omega(x), z - P_\Omega(x)\rangle \leqslant 0, \quad \text{for all } z \in \Omega \tag{54}$$

Equation (54) is called the ***minimum principle***.

Remark:

- Minimum principle characterizes the projection $P_\Omega(x)$, that is, no other point $\bar{x} \in \Omega$ satisfies the minimum principle.
- This comes from the uniqueness of the projection operator.

7.23.1 Fixed Point of Projected Gradient Descent (PGD)

Lemma 7.33. *Let f be M-smooth and Ω be closed and convex. Then any $x^* \in \Omega$ satisfies*

$$-\nabla f(x^*) \in N_\Omega(x^*) \quad \text{if and only if} \quad x^* = P_\Omega(x^* - \alpha \nabla f(x^*)) \tag{55}$$

Proof: "⇒"
Suppose x^* satisfies the first-order necessary condition. Suppose $\alpha > 0$ is the stepsize.

$$\begin{aligned} &\Rightarrow \quad \langle -\nabla f(x^*), z - x^* \rangle \quad \leqslant 0 \\ &\Rightarrow \quad \langle -\alpha \nabla f(x^*), z - x^* \rangle \quad \leqslant 0 \\ \Rightarrow \quad &\langle x^* - \alpha \nabla f(x^*) - x^*, z - x^* \rangle \quad \leqslant 0, \quad \forall z \in \Omega \end{aligned}$$

So, from the minimum principle,

$$x^* = P_\Omega(x^* - \alpha \nabla f(x^*))$$

"⇐"
We have $x^* = P_\Omega(x^* - \alpha \nabla f(x^*))$.
Invoking minimum principle, we get

$$\begin{aligned} &\langle x^* - \alpha \nabla f(x^*) - P_\Omega(x^* - \alpha \nabla f(x^*)), z - P_\Omega(x^* - \alpha \nabla f(x^*)) \rangle \quad \leqslant 0 \\ &\Rightarrow \quad \langle -\alpha \nabla f(x^*), z - x^* \rangle \quad \leqslant 0 \\ &\Rightarrow \quad \langle -\nabla f(x^*), z - x^* \rangle \quad \leqslant 0 \\ \Rightarrow \quad &-\nabla f(x^*) \in N_\Omega(x^*) \quad \textit{(by definition of normal cone)} \end{aligned}$$

□

7.23.2 Contraction/ Non-expansive Property of Projection

Suppose Ω is a closed convex set. Then

$$\|P_\Omega(x) - P_\Omega(y)\| \leqslant \|x - y\|, \quad \forall x, y \in \mathbb{R}^n \tag{56}$$

Proof: We have

$$\begin{aligned} &\|x - y\|^2 \\ &= \|(x - P_\Omega(x)) - (y - P_\Omega(y)) + (P_\Omega(x) - P_\Omega(y))\|^2 \\ &= \|(x - P_\Omega(x)) - (y - P_\Omega(y))\|^2 + \|(P_\Omega(x) - P_\Omega(y))\|^2 \\ &- 2\langle x - P_\Omega(x), P_\Omega(y) - P_\Omega(x) \rangle \quad (\geqslant 0 \text{ by minimum principle}) \\ &- 2\langle y - P_\Omega(y), P_\Omega(x) - P_\Omega(y) \rangle \quad (\geqslant 0 \text{ by minimum principle}) \\ &\geqslant \|(x - P_\Omega(x)) - (y - P_\Omega(y))\|^2 + \|P_\Omega(x) - P_\Omega(y)\|^2 \\ &\geqslant \|P_\Omega(x) - P_\Omega(y)\|^2, \end{aligned}$$

$$\Rightarrow \|x-y\| \geqslant \|P_\Omega(x) - P_\Omega(y)\| \tag{57}$$

□

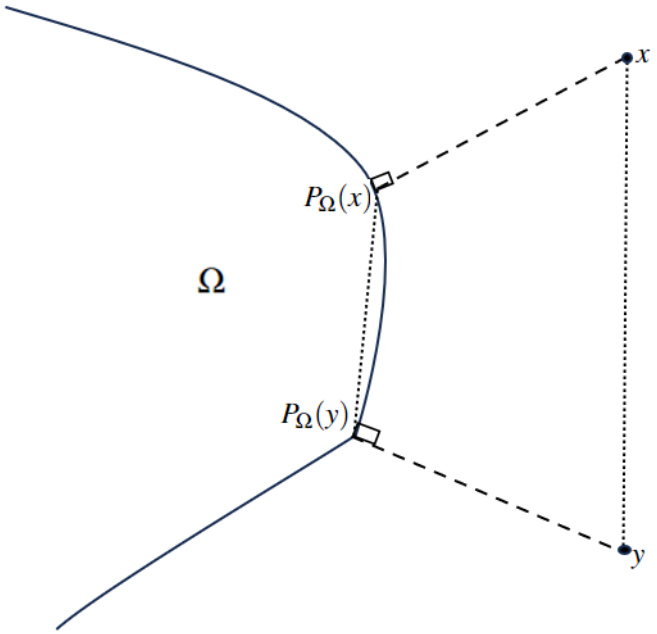


Figure 7.7. Euclidean Projection of points x and y on Ω

Figure 7.1 shows the non-expansive property of Euclidean projection. The distance between projection points $P_\Omega(x)$ and $P_\Omega(y)$ is always less than or equal to the distance between x and y.

7.23.3 Convergence of PGD on Smooth and Convex Objectives

(Reference: Optimization for Data Analysis, Stephen Wright and Benjamin Recht, Chapter 7 [9])

Theorem 7.34. *Suppose f is convex and M-smooth. We run PGD with $\alpha^k = 1/M$ for T iterations. We obtain*

$$f(x^T) - f(x^*) \leqslant \frac{M}{2T} \left\|x^0 - x^*\right\|^2 \tag{58}$$

Proof: Denote the update

$$T(x) = P_\Omega \left[x - \frac{1}{M}\nabla f(x)\right] \tag{59}$$

Define the residual

$$g_\Omega(x) = M\left[x - T(x)\right] \tag{60}$$

With this,

$$x^{k+1} = T(x^k)$$

$$\text{and } x^{k+1} = x^k - \frac{1}{M} g_\Omega(x^k) \tag{61}$$

Lemma 7.35. ***(Descent Lemma)*** *For all $x, y \in \Omega$,*

$$f(T(x)) - f(y) \leqslant \langle g_\Omega(x), x - y \rangle - \frac{1}{2M} \|g_\Omega(x)\|^2 \tag{62}$$

We take this lemma as given and conclude the theorem proof first.

Plug $x = y = x^k$ and $T(x^k) = x^{k+1}$ in the lemma.
We get the usual Descent Lemma

$$f(x^{k+1}) - f(x^k) \leqslant -\frac{1}{2M} \left\| g_\Omega(x^k) \right\|^2 \tag{63}$$

Now, we take $x = x^k$, $y = x^*$ and use Descent Lemma.

$$\begin{aligned}
f(x^{k+1}) - f(x^*) \leqslant & \left\langle g_\Omega(x^k), x^k - x^* \right\rangle - \frac{1}{2M} \left\| g_\Omega(x^k) \right\|^2 \\
= & \frac{M}{2} \left[\left\| x^k - x^* \right\|^2 - \left\| x^k - \frac{1}{M} g_\Omega(x^k) - x^* \right\|^2 \right] \qquad \text{(Completing the squares)} \\
= & \frac{M}{2} \left[\left\| x^k - x^* \right\|^2 - \left\| x^{k+1} - x^* \right\|^2 \right] \qquad \text{(Using (61))}
\end{aligned}$$

Now taking sum to T, and using telescopic sum for RHS,

$$\begin{aligned}
f(x^T) - f(x^*) & \leqslant \frac{1}{T} \sum_{k=1}^{T} f(x^k) - f(x^*) \qquad \text{(Using (63))} \\
& \leqslant \frac{M}{2T} \sum_{k=1}^{T} \left[\left\| x^{k-1} - x^* \right\|^2 - \left\| x^k - x^* \right\|^2 \right] \\
& = \frac{M}{2T} \left[\left\| x^0 - x^* \right\|^2 - \left\| x^T - x^* \right\|^2 \right] \\
& \leqslant \frac{M}{2T} \left\| x^0 - x^* \right\|^2
\end{aligned}$$

□

Proof of Descent Lemma:

Proof: Define

$$\begin{aligned}
S(x) &= x - \frac{1}{M} \nabla f(x) \\
T(x) &= P_\Omega(S(x))
\end{aligned} \tag{64}$$

Using the minimum principle,

$$\begin{aligned}
\langle S(x) - P_\Omega(S(x)), y - P_\Omega(S(x)) \rangle & \leqslant 0, \quad \forall y \in \Omega \\
\langle S(x) - T(x), y - T(x) \rangle & \leqslant 0 \\
T(x) = x - \frac{1}{M} g_\Omega(x) \quad ; \quad S(x) &= x - \frac{1}{M} \nabla f(x) \\
\langle \nabla f(x), T(x) - y \rangle & \leqslant \langle g_\Omega(x), T(x) - y \rangle
\end{aligned} \tag{65}$$

LHS of Lemma:

$$\begin{aligned}
f(T(x)) - f(y) &= [f(T(x)) - f(x)] + [f(x) - f(y)] \\
&\leqslant \langle \nabla f(x), T(x) - x \rangle + \frac{M}{2} \|T(x) - x\|^2 + \langle \nabla f(x), x - y \rangle \\
&\qquad\qquad \text{(Using smoothness and convexity of } f\text{)} \\
&= \langle \nabla f(x), T(x) - y \rangle + \frac{1}{2M} \|g_\Omega(x)\|^2 \\
&\leqslant \langle g_\Omega(x), T(x) - y \rangle + \frac{1}{2M} \|g_\Omega(x)\|^2 \qquad \text{(Using (65))} \\
&= \langle g_\Omega(x), x - y \rangle + \langle g_\Omega(x), T(x) - x \rangle + \frac{1}{2M} \|g_\Omega(x)\|^2 \\
&= \langle g_\Omega(x), x - y \rangle - \frac{1}{2M} \|g_\Omega(x)\|^2
\end{aligned}$$

□

7.23.4 PGD for M-Smooth and m-Strongly Convex Objective

Theorem 7.36. *Choose $\alpha^k = \frac{2}{M+m}$. We have*

$$\left\| x^{k+1} - x^* \right\|_2 \leqslant \left(\frac{1 - m/M}{1 + m/M} \right)^k \left\| x^0 - x^k \right\|_2 \tag{66}$$

NOTE: We have given guarantees of 3 types till now:

1. $\left\| \nabla f(x^k) \right\|$
2. $f(x^k) - f(x^*) \Rightarrow$ Stepsize : $1/M$
3. $\left\| x^k - x^* \right\| \Rightarrow$ Stepsize : $2/(M+m)$

Objective function:

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

Projected Gradient descent is one algorithm for constrained optimization, we require

$$P_\Omega(x) = \arg\min_{z\in\Omega} \frac{1}{2}\|x-z\|^2.$$

Examples:

1. Non-negative orthant
2. Box Constraints $a_j \leqslant x_j \leqslant b_j$
3. l_2 Norm Ball
4. l_1 Norm Ball (Soft Theresholding function) (Note: Refer Lecture 6)
5. Postive Semi-Definite Cone $\{x \in \mathbb{R}^{d\times d} | x = x^T, x \succcurlyeq 0\}$
6. Nuclear norm ball $\{x \in \mathbb{R}^{d_1\times d_2} | \sum_{j=1}^m \sigma_j(x) \leqslant 1, m = \min\{d_1,d_2\}\}$. Where σ represents the singular values of x.

8.24 Frank Wolfe/Conditional Gradient

This method aims at avoiding projection. Let Ω be a closed, convex, and bounded set.

At iteration k (x^k current state). Take the linearized approximation of the function of around x^k

$$\bar{x}^k = \arg\min_{y\in\Omega} f(x^k) + \left\langle \nabla f(x^k), y - x^k \right\rangle = \arg\min_{y\in\Omega} \left\langle \nabla f(x^k), y \right\rangle \tag{67}$$

$\bar{x}^k$ is called the Frank Wolfe direction.

The new iterate is given by

$$\begin{aligned} x^{k+1} &= (1-\alpha^k)x^k + \alpha^k \bar{x}^k \\ &= x^k - \alpha^k(x^k - \bar{x}^k)\ , \alpha^k \in [0,1] \end{aligned} \tag{68}$$

(Note: Multiple $\bar{x}^k$ may exist due to the underlying assumption of convexity and not strong convexity)

Example 8.24.1. *Consider the l_2-norm ball $\{x \in \mathbb{R}^d | \|x\|_2 \leqslant 1\}$.*

$$\begin{aligned} \bar{x}^k &= \arg\min_{\|y\|_2\leqslant 1} \left\langle \nabla f(x^k), y \right\rangle \textit{ (refer dual norm)} \\ \bar{x}^k &= \frac{-\nabla f(x^k)}{\left\|\nabla f(x^k)\right\|} \end{aligned}$$

Example 8.24.2. *Consider the l_1-norm ball $\{x \in \mathbb{R}^d | \|x\|_1 \leqslant 1\}$.*

$$\bar{x}^k = \arg\min_{\|y\|_1 \leqslant 1} \left\langle \nabla f(x^k), y \right\rangle = -e_j \; sign(\nabla f(x^k)_j),$$

where $e_j = \begin{pmatrix} 0 & \cdots & 1 & \cdots & 0 \end{pmatrix}^\top$.

8.25 Convergence guarantees of Frank Wolfe

Lemma 8.37. *The iterates $\{x^k\}_{k=1} \in \Omega$ if $x_0 \in \Omega$*

Proof: Observe that

$$\begin{aligned} x^{k+1} &= x^k - \alpha^k(x^k - \bar{x}^k) \text{ From (68)} \\ \bar{x}^k &= \arg\min_{y \in \Omega} \left\langle \nabla f(x^k), y \right\rangle \text{ From (67)} \\ \bar{x}^k &\in \Omega \text{ as } \alpha^k \in [0,1] \\ x^{k+1} &\in \Omega \end{aligned}$$

□

Theorem 8.38. *Suppose Ω is convex, closed and bounded with diameter D. $f(.)$ is M smooth and convex. $\alpha^k = \frac{2}{k+2}$. Frank Wolfe algorithm yields*

$$f(x^k) - f(x^*) \leqslant \frac{2MD^2}{k+2} \quad \text{for } k = 1, 2..., $$

where D is the diameter which is given $D = \max_{x,y \in \Omega} \|x - y\|$.

Remark 8.39. *The stepsize is diminishing*

Remark 8.40. *convergence rate = $O(\frac{1}{iteration})$*

Proof: Since $f(\cdot)$ is M smooth

$$\begin{aligned} f(x^{k+1}) &\leqslant f(x^k) + \left\langle \nabla f(x^k), x^{k+1} - x^k \right\rangle + \frac{M \left\| x^{k+1} - x^k \right\|^2}{2} \\ x^{k+1} &= x^k - \alpha^k(x^k - \bar{x}^k) \text{ (From (68))} \\ f(x^{k+1}) &\leqslant f(x^k) + \left\langle \nabla f(x^k), \alpha^k(x^k - \bar{x}^k) \right\rangle + \frac{M \left\| \alpha^k(x^k - \bar{x}^k) \right\|^2}{2} \\ f(x^{k+1}) &\leqslant f(x^k) + \alpha^k \left\langle \nabla f(x^k), (x^k - \bar{x}^k) \right\rangle + \frac{M(\alpha^k D)^2}{2} \qquad (69) \\ \left\langle \nabla f(x^k), \bar{x}^k \right\rangle &\leqslant \left\langle \nabla f(x^k), \bar{x}^* \right\rangle \text{ (since } \bar{x}^k \text{ if FW direction)} \\ \alpha^k \left\langle \nabla f(x^k), \bar{x}^k - x^k \right\rangle &\leqslant \alpha^k \left\langle \nabla f(x^k), \bar{x}^* - x^k \right\rangle \\ \alpha^k \left\langle \nabla f(x^k), \bar{x}^k - x^k \right\rangle &\leqslant \alpha^k [f(x^*) - f(x^k)] \text{ (since } f(.) \text{ is Convex)} \qquad (70) \\ f(x^{k+1}) &\leqslant f(x^k) + \alpha^k [f(x^*) - f(x^k)] + \frac{M(\alpha^k D)^2}{2} \text{ (From (69),(70))} \\ f(x^{k+1}) - f(x^*) &\leqslant (1 - \alpha^k)[f(x^k) - f(x^*)] + \frac{M(\alpha^k D)^2}{2} \end{aligned}$$

Base Case $k = 0, \alpha^0 = 1$

$$f(x^1) - f(x^*) \leqslant \frac{MD^2}{2} \leqslant \frac{2MD^2}{3} \text{ (Theorem is valid for 1st iteration)}$$

Case with some non-zero k iteration

$$\begin{aligned}
f(x^{k+1}) - f(x^*) &\leqslant (1-\alpha^k)[f(x^k) - f(x^*)] + \frac{M(\alpha^k D)^2}{2} \\
f(x^{k+1}) - f(x^*) &\leqslant (1-\frac{2}{k+2})[f(x^k) - f(x^*)] + \frac{M(\frac{2D}{k+2})^2}{2} \\
f(x^{k+1}) - f(x^*) &\leqslant \frac{k}{k+2}\frac{2MD^2}{k+2} + \frac{2MD^2}{(k+2)^2} \\
f(x^{k+1}) - f(x^*) &\leqslant \frac{2MD^2(k+1)}{(k+2)^2} \\
f(x^{k+1}) - f(x^*) &\leqslant \frac{2MD^2(k+2)}{(k+2)(k+3)} \\
f(x^{k+1}) - f(x^*) &\leqslant \frac{2MD^2}{(k+3)}
\end{aligned}$$

□

Lecture 9 — September 1

Lecturer: Avishek Ghosh | *Scribe: Yashvardhan*

9.26 SubGradient Methods: Introduction

We want to solve $\min_{x\in\Omega} f(x)$ where f is convex but non-differentiable (non-smooth).

Example 9.26.1. $f(x) = \max_{j=1,2,3} \{\langle a_j, x\rangle + b_j \mid \forall j, a_j \in \mathbb{R}^d, b_j \in \mathbb{R}\}$.

Example 9.26.2. $f(x) = |x|$.

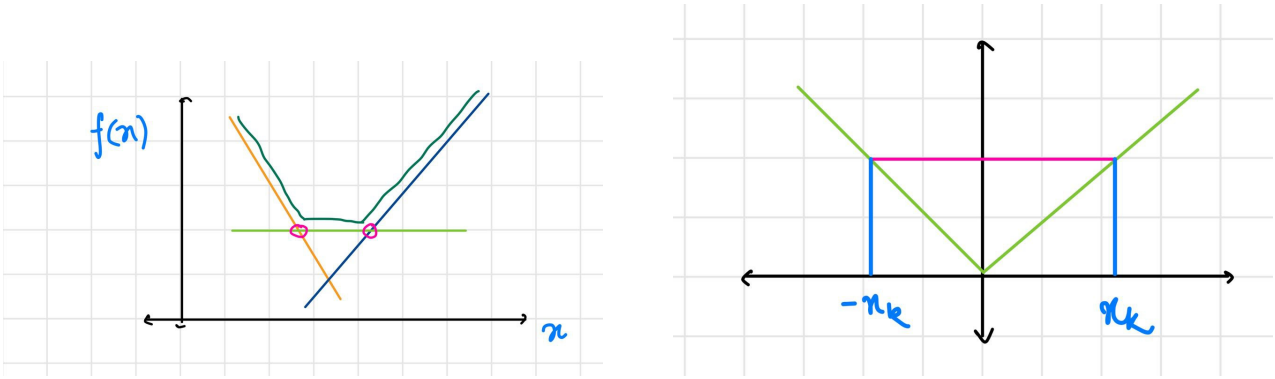


Figure 9.8. peice-wise and mod function

9.27 Geometric Intuition of Gradients

Definition 9.41 (Epigraph). The epigraph or supergraph of a function f valued in the extended real numbers is the set, denoted by of all points in the Cartesian product lying on or above its graph i.e.

$$epi(f) = \{(x,t) \in \mathbb{R}^d \times \mathbb{R} \mid f(x) \leqslant t\} \tag{71}$$

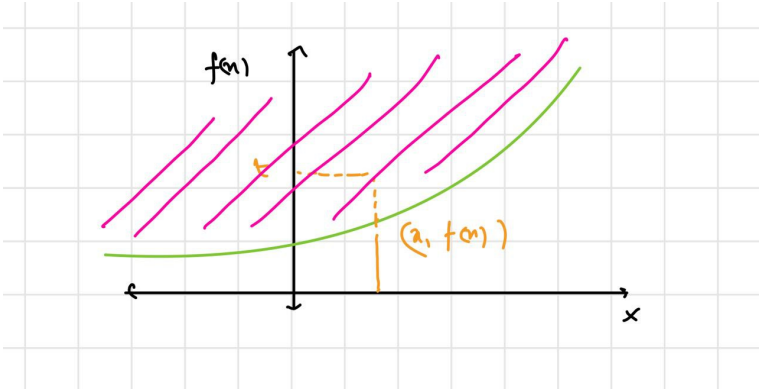


Figure 9.9. Pink part marks the epi-graph of f

Exercise 9.42. *Prove function f:* $\mathbb{R}^d \to \mathbb{R}$ *is convex if and only if* $epif(f)$ *is a convex set.*

We allow f to take extended real values such that,

$$f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$$
$$dom(f) = \{x \in \mathbb{R}^d \mid f(x) < +\infty\}$$

Remark 9.43. Tangent-line at a point x is supporting hyperplane to the $epi(f)$.

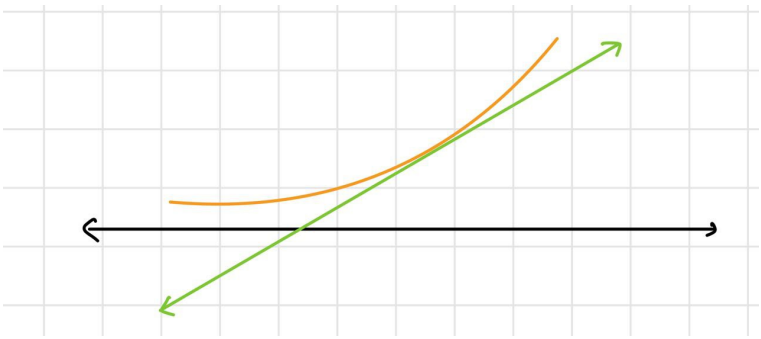


Figure 9.10. Orange is function f and green the tangent to point f at point x

Definition 9.44 (Sub-Gradient). We say a vector $g \in \mathbb{R}^{\times}$ is a sub-gradient of $f : \mathbb{R}^d \to \mathbb{R}$ at $x \in dom(f)$ if $\forall y \in dom(f)$,

$$f(y) \geqslant f(x) + \langle g, (y - x) \rangle, \text{ we write } g \in \partial f(x) \text{ as set of sub-gradients.}$$

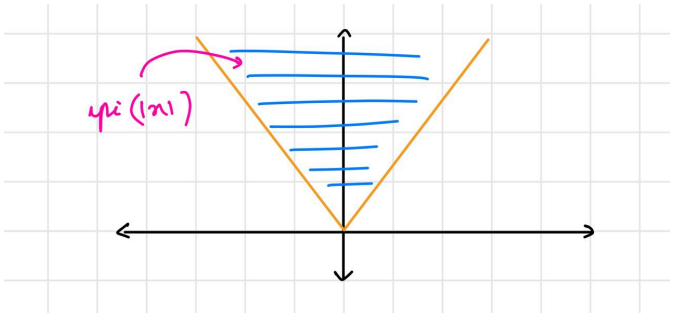
Exercise 9.45. *Prove* f *is differentiable at* $x \iff \partial f(x) = \{\nabla f(x)\}$.

Example 9.27.1. $f(x) = |x|$, *then sub-gradient* $\partial f(x)$.

9.28 Connection to Optimization

Let x^* be the solution of optimization problem,

$$\min_{x \in \mathbb{R}^d} \{f(x)\} \text{ where } f \text{ is not differentiable.}$$



$$\partial f(x) = \begin{cases} \{+1\} & x > 0 \\ \{-1\} & x < 0 \\ [-1,+1] & x = 0 \end{cases}$$

Table 9.1. sub-gradient for $|x|$

x^* is global mimina $\iff 0 \in \partial f(x^*)$. If f is differentiable at x^* then $\partial f(x^*) = \{\nabla f(x^*)\}$ and $\nabla f(x^*) = 0$.

Proof: Forward: $0 \in \partial f(x^*)$

$$\implies f(y) \geqslant f(x^*) + \langle 0, y - x^* \rangle \ \forall y \in dom(f)$$
$$\implies f(y) \geqslant f(x^*)$$

Thus x^* is Global Minima.
Backward: given x^* is global minima,

$$\implies f(y) \geqslant f(x^*)$$
$$\implies f(y) \geqslant f(x^*) + \langle 0, y - x^* \rangle \quad \forall y \in dom(f)$$

Hence $0 \in \partial f(x^*)$. □

Example 9.28.1. *Indicator Function* Let Ω is closed convex set.

$$\mathbb{I}_\Omega(x) = \begin{cases} 0 & x \in \Omega \\ +\infty & otherwise \end{cases}$$

This is a convex function (Extended values).
Sub Gradient at a point x be $g \in \partial \mathbb{I}_\Omega(x)$ such that,

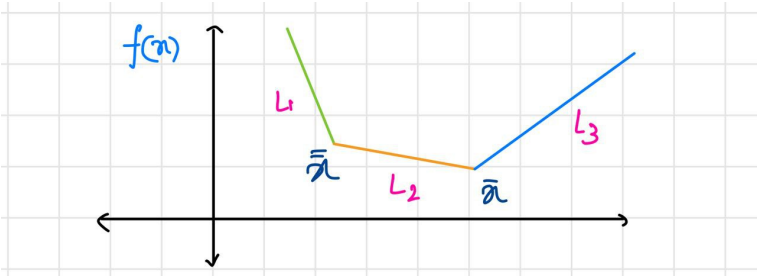
$$\mathbb{I}_\Omega(y) \geqslant \mathbb{I}_\Omega(x) + \langle g, y - x \rangle \forall y$$

Case 1 If $y \notin \Omega : \mathbb{I}_\Omega(y) = +\infty$ hence any $g \in \mathbb{R}^d$ satisfies sub-gradient condition.
Case 2 If $y \in \Omega$, we will be looking at x, where $x \in \Omega$,

$$0 \geqslant 0 + \langle g, y - x \rangle$$
$$\langle g, y - x \rangle \leqslant 0 \quad \forall y$$
$$\text{which implies } g \in \mathcal{N}_\Omega(x)$$
$$\text{hence} \quad \partial \mathbb{I}_\Omega(x) = \mathcal{N}_\Omega(x).$$

Example 9.28.2. ***Piece-wise Linear Function***

$$f(x) = \max_{j=1,2,3} \{\langle a_j, x \rangle + b_j\} \quad \forall j, a_j \in \mathbb{R}^d, b_j \in \mathbb{R}$$



Define $M(x) = \{j = 1,2,3 \mid f(x) = \langle a_j, x\rangle + b_j\}$.
$\partial f(x) =$ convex hull $\{a_k \mid k \in M(x)\}$. Example,

$$M(\bar{\bar{x}}) = \{1,2\} \quad \partial f(\bar{\bar{x}}) = \{\lambda a_1 + (1-\lambda)a_2 | \lambda \in [0,1]\} \tag{72}$$
$$M(\bar{x}) = \{2,3\} \quad \partial f(\bar{x}) = \{\lambda a_2 + (1-\lambda)a_3 | \lambda \in [0,1]\} \tag{73}$$

9.29 Connection to the Constrained Problem

Let x^* be the solution of contained problem,

$$\min_{x\in\Omega} f(x) \text{ or}$$
$$\min_{x\in\mathbb{R}^d} \{f(x) + \mathbb{I}_\Omega(x)\}$$

From proposition 9.28

$$0 \in \partial(f + \mathbb{I}_\Omega)(x^*)$$

which under some condition is equivalent to

$$0 \in \partial f(x^*) + \mathbb{I}_\Omega(x^*)$$

for differentiable

$$0 \in \nabla f(x^*) + \mathcal{N}_\Omega(x^*)$$
$$\implies -\nabla f(x^*) \in \mathcal{N}_\Omega(x^*)$$

10.30 Last lecture

In the last lecture, we studied the geometric ideas related to sub-gradients and their relation to optimization.

In the current lecture, we will continue with several important properties of sub-gradients. We will study an optimization algorithm based on sub-gradients.

10.31 Calculus for Sub-Gradients

We start with two important properties of sub-gradients.

Theorem 10.46. *Scaling theorem for sub-gradients*

$$\partial(\lambda f(x)) = \lambda \partial f(x) \ , \lambda \geqslant 0 \in \mathbb{R}.$$

Proof: By the definition of Sub-Gradients for $x, y \in \mathbb{R}^d$ and $g \in \partial f(x)$

$$f(y) \geqslant f(x) + \langle g, y - x \rangle.$$

Multiplying both sides of the inequality by λ

$$\lambda f(y) \geqslant \lambda f(x) + \lambda \langle g, y - x \rangle.$$

Let $\lambda f(x) = h(x)$ and

$$h(y) \geqslant h(x) + \langle \lambda g, y - x \rangle.$$

Thus we can claim that $\lambda g \in \partial h(x)$. Since, $g \in \partial f(x)$. We get,

$$\lambda g \in \partial f(x) \implies \lambda \partial f(x) = \partial h(x) \implies \lambda \partial f(x) = \partial(\lambda f(x)).$$

□

Theorem 10.47. *Addition Theorem for Sub-Gradients*

$$\partial f_1(x) + \partial f_2(x) \subseteq \partial(f_1 + f_2)(x).$$

Proof: Define $g_1 \in \partial f_1(x), g_2 \in \partial f_2(x)$ and we claim that ,

$$g_1 + g_2 \in \partial(f_1 + f_2)(x).$$

From Definition of Sub-gradients, $\forall\ y \in \mathbb{R}$

$$f_1(y) \geqslant f_1(x) + \langle g_1, y - x \rangle$$

$$f_2(y) \geqslant f_2(x) + \langle g_2, y-x \rangle.$$

Adding both inequalities,

$$(f_1+f_2)(y) \geqslant (f_1+f_2)(x) + \langle g_1, y-x \rangle + \langle g_2, y-x \rangle$$

$$(f_1+f_2)(y) \geqslant (f_1+f_2)(x) + \langle g_1+g_2, y-x \rangle$$

Thus, $g_1+g_2 \in \partial(f_1+f_2)(x)$.

Now we prove the converse for the Addition Theorem, that is the reverse expression $\partial(f_1+f_2)(x) \neq \partial f_1(x) + \partial f_2(x)$ is not true in general. Recall that the indicator function $1_Q(x)$ for a given closed bounded set $\mathbb{Q}$ is defined by

$$1_Q(x) = \begin{cases} 0 & \text{if } x \in \mathbb{Q} \\ \infty & \text{if } x \notin \mathbb{Q}. \end{cases}$$

Define the set $\Omega_- = \{x \in \mathbb{R}^2 : \|x-(-1,0)\| \leqslant 1\}$. Note that $\Omega_+ = \{x \in \mathbb{R}^2 : \|x-(1,0)\| \leqslant 1\}$ and $\Omega_- \cap \Omega_+ = \{(0,0)\}$. Then we write

$$f_1(x) = I_{\Omega_-}(x), f_2(x) = I_{\Omega_+}(x)$$

$$(f_1+f_2)(x) = I_{\Omega_- \cap \Omega_+}(x)$$

Fix any vector $g \in \mathbb{R}^2$. We claim that $g \in \partial(f_1+f_2)(x)$. Using the Definition of Sub-Gradients,

$$(f_1+f_2)(y) \geqslant (f_1+f_2)(x) + \langle g, y-x \rangle.$$

Since the expression is undefined for $x, y \in \mathbb{R}^2 - \{(0,0)\}$, we conclude that

$$y = x \implies \langle g, 0 \rangle = 0 \implies g \in \mathbb{R}^2.$$

Our proof is complete. □

10.32 Algorithms Related to Sub-Gradient

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{X}} f(x)$$

where $f(x)$ is non-differentiable and non-smooth. We choose $g^k \in \partial f(x^k)$

$$x^{k+1} = x^k - \alpha^k g^k, \alpha^k \geqslant 0$$

Assumption : g is G Lipschitz $\|g\| \leqslant G$ for any $g \in f(x)$ $\forall x \in \mathbb{R}$.

Theorem 10.48. *Bounds of the unconstrained optimization problem with a non-differential objective using standard gradient descent.*

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{\|x^1 - x^*\|^2 + G^2 \sum_{k=1}^{T} (\alpha^k)^2}{2 \sum_{k=1}^{T} \alpha^k}$$

where f is convex and $\bar{x} = \dfrac{\sum_{k=1}^{T} \alpha^k x^k}{\sum_{k=1}^{T} \alpha^k}$ *and* $D_0 = \|x^1 - x^*\|$.

Proof: Some basic observations if $\alpha^k = c \in \mathbb{R}$,

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{D_0^2 + G^2 T c^2}{2Tc} = \frac{D_0^2}{2Tc} + \frac{G^2 c}{2}.$$

Our choice of α^k should be such that it satisfies the following properties

$$\sum_{k=1}^{T} \alpha^k = \infty, \sum_{k=1}^{T} (\alpha^k)^2 = 0,$$
$$g^k = \partial f(x^k).$$

Starting with the basic gradient descent iterate k

$$x^{k+1} = x^k - \alpha^k g^k, \alpha^k \geqslant 0,$$
$$x^{k+1} - x^* = x^k - x^* - \alpha^k g^k.$$

Squaring both sides

$$\|x^{k+1} - x^*\|^2 \leqslant \|x^k - x^*\|^2 - 2\alpha^k \langle g^k, x^k - x^* \rangle + (\alpha^k)^2 G^2.$$

Thus,

$$\alpha^k \langle g^k, x^k - x^* \rangle \leqslant \frac{1}{2}(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2) + (\alpha^k)^2 G^2,$$
$$f(y) \geqslant f(x^k) + \langle g^k, y - x^k \rangle.$$

Put $y = x^*$,

$$f(x^*) \geqslant f(x^k) + \langle g^k, x^* - x^k \rangle,$$
$$\alpha^k (f(x^k) - f(x^*)) \leqslant \alpha^k \langle g^k, x^* - x^k \rangle \leqslant \frac{1}{2}(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 + (\alpha^k)^2 G^2).$$

Now with index as k summing from k=1 to k=T and dividing by $\sum_{k=1}^{T} \alpha^k$,

$$\frac{\sum_{k=1}^{T} \alpha^k (f(x^k) - f(x^*))}{\sum_{k=1}^{T} \alpha^k} \leqslant \frac{\sum_{k=1}^{T} \frac{1}{2}(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2)}{\sum_{k=1}^{T} \alpha^k}.$$

Now we deal with terms individually , Weighted mean of a iterate is lower bounded by normal mean,(Jensen's Inequality)

$$\frac{\sum_{k=1}^{T} \alpha^k f(x^k)}{\sum_{k=1}^{T} \alpha^k} \geqslant f(\bar{x}^T)$$

Consider the telescoping sum

$$\sum_{k=1}^{T}(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2) = \|x^1 - x^*\|^2 - \|x^2 - x^*\|^2 + \ldots \|x^T - x^*\|^2 - \|x^{T+1} - x^*\|^2$$

$$\sum_{k=1}^{T}(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2) = \|x^1 - x^*\|^2 - \|x^{T+1} - x^*\|^2$$

Our original inequality,

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{\sum_{k=1}^{T}\alpha^k(f(x^k) - f(x^*))}{\sum_{k=1}^{T}\alpha^k} \leqslant \frac{\sum_{k=1}^{T}\frac{1}{2}(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 + (\alpha^k)^2G^2)}{\sum_{k=1}^{T}\alpha^k}$$

$$\leqslant \frac{\|x^1 - x^*\|^2 - \|x^{T+1} - x^*\|^2 + \sum_{k=1}^{T}(\alpha^k)^2G^2}{2\sum_{k=1}^{T}\alpha^k}$$

$$\leqslant \frac{\|x^1 - x^*\|^2 + \sum_{k=1}^{T}(\alpha^k)^2G^2}{2\sum_{k=1}^{T}\alpha^k}$$

Define $D_0^2 = \|x^1 - x^*\|^2$ Hence,

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{D_0^2 + \sum_{k=1}^{T}(\alpha^k)^2G^2}{2\sum_{k=1}^{T}\alpha^k}$$

□

10.32.1 Oracle Complexity for non-differential objective using sub- gradients

1. Case 1: $\alpha^k = \alpha$ is fixed but small
   Since,

   $$f(\bar{x}^T) - f(x^*) \leqslant \frac{D_0^2 + \sum_{k=1}^{T}(\alpha^k)^2G^2}{2\sum_{k=1}^{T}\alpha^k}$$

   Evaluates to

   $$f(\bar{x}^T) - f(x^*) \leqslant \frac{D_0^2 + \alpha^2G^2}{2\alpha}$$

Define $h(\alpha) = \frac{D_0^2+\alpha^2G^2}{2\alpha}$ For α_{min} make $h'(\alpha) = 0$

$$-\frac{D_0^2}{2T\alpha^2} + \frac{G^2}{2} = 0$$

$$\alpha = \frac{D_0}{G\sqrt{T}}$$

Thus the time complexity for a given α, D_0, T is $O(\frac{1}{\sqrt{T}})$

2. Case 2: $\alpha^k = \frac{C}{\sqrt{K}}, C > 0$
   Again,

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{D_0^2 + \sum_{k=1}^{T} (\alpha^k)^2 G^2}{2\sum_{k=1}^{T} \alpha^k}$$

   Evaluates to

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{D_0^2 + \sum_{k=1}^{T} (\frac{1}{k}) C^2 G^2}{2C\sum_{k=1}^{T} \frac{1}{\sqrt{k}}}$$

   We know $0 < k \leqslant T \ \forall k \implies \sum_{k=1}^{T} \frac{1}{\sqrt{T}} \leqslant \sum_{k=1}^{T} \frac{1}{\sqrt{k}}$

   Also $\sum_{k=1}^{T} \frac{1}{T} \leqslant 1 + \log(T)$ by the property of Harmonic Sums
   Thus Original inequality evaluates to ,

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{D_0^2 + (1+\log(T) C^2 G^2}{2C\sqrt{T}}$$

   Thus the oracle time complexity is $O(\frac{\log T}{\sqrt{T}})$.

Remark 10.49. *Time complexity of Case-2 is worse than that of Case-1 but it requires the initialization of lesser parameters (e.g. D_0) than for the case of fixed α^k. Therefore it is a trade-off.*

Lecture 11 — September 8

Lecturer: Avishek Ghosh *Scribe: Ayush Srivastava*

11.33 Recap

We have been using subgradient methods for solving:

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

f being non-differentiable/non-smooth

(a) First, we defined subgradient (whose existence is guaranteed by the Supporting Hyperplane Theorem).

(b) Subgradient descent method: $x^{k+1} = x^k - (\alpha)^k g^k, k = 1,2,\ldots$; for $g^k \in \partial f(x^k)$

We want $\min f(x)$, where $x \in \mathbb{R}^d$ and f is G-Lipschitz and convex. Let x_0 be the initial point such that $\|x_0 - x_*\| \leqslant D$; and x_* be the minima.
We restrict ourselves to the following class of 1st order algorithms:
At round k:

- Algorithm receives $g^k \in \partial f(x^k)$
- Algorithm generates next iterate $x^{k+1} \in x^0 + \text{span}\{g^0,\ldots,g^k\}$

Example - Usual subgradient descent

Theorem 11.50. *For any such 'first-order' algorithm, with initialization x^0, there exists a function f (convex and Lipschitz), such that for any iteration $k \leqslant d-1$,*

$$f(x^k) - f(x^*) \geqslant \frac{DG}{2(1+\sqrt{k+1})}$$

Proof: Without loss of generality, assume $x^0 = 0$
We can recenter our function f WLOG. We will prove this claim for $k = d-1$ and see that $k \leqslant d-1$ will follow similarly.

- **Step 1:** Function construction:
  $f(x) = \max_{j=1,\ldots,d} x_j + \frac{1}{2}\|x\|_2^2$

  We will look at its behaviour over l_2-ball with radius D
  For any x, y such that $\|x\| \leqslant D$
  We have

$$\begin{aligned} \text{mod } f(x) - f(y) &\leqslant \|x-y\|_\infty + \frac{1}{2}(\|x\| - \|y\|)(\|x\| + \|y\|) \\ &\leqslant \|x-y\|_2 + D\|x-y\|_2 \\ &= (1+D)\|x-y\|_2 \\ &= (1+\frac{1}{\sqrt{d}})\|x-y\|_2 \end{aligned}$$

We choose $D = \frac{1}{\sqrt{d}}$
Thus, f is $(1 + \frac{1}{\sqrt{d}})$-Lipschitz

- **Step 2:** (Example)

$$x^* = -\frac{1}{d}(1, 1, \ldots, 1)$$
$$\implies f(x^*) = -\frac{1}{2d} \leqslant 0$$

- **Step 3:** Suppose our algorithm outputs the subgradients at point x

$$\begin{aligned} g &= e_J + x \\ J &= \min(j = 1, \ldots, d | x_j = \max_{k=1,\ldots,d} x_k) \\ \partial f(x) &= x + \partial(\max_{j=1,\ldots,d} \langle e_j, x \rangle) \\ &= x + \text{cvxhull}(e_j | x_j = \max_{k=1,\ldots,d} x_k) \end{aligned}$$

So, $g \in \partial f(x)$. Now, by construction, $x^0 = 0$, and $g^0 = e_1$

$$\begin{aligned} x^1 &\in span\{e_1\} \\ x^2 &\in span\{e_1, e_2\} \\ x^{d-1} &\in span\{e_1, \ldots, e_{d-1}\} \end{aligned}$$

Then, at $k = d - 1$, we have

$$[x^{d-1}]_d = 0$$

Now,

$$\max_{j=1,\ldots,d} x_j^{d-1} \geqslant 0$$

So, we have

$$f(x^{d-1}) - f(x^*) \geqslant \frac{1}{2d}$$

For $D = \frac{1}{\sqrt{d}}$, and $G = 1 + \frac{1}{d}$, we have $DG = \frac{1}{d}(1 + \sqrt{d})$
$\implies d = \frac{1+\sqrt{d}}{DG}$
Substituting, we get the following

$$\begin{aligned} f(x^{d-1}) - f(x^*) &\geqslant \frac{DG}{2(1 + \sqrt{d})} \\ \implies f(x^{d-1}) - f(x^*) &\geqslant \frac{DG}{2(1 + \sqrt{(d-1)+1})} \end{aligned}$$

We have shown for $k = d - 1$. The same argument holds for $k \leqslant d - 2$

□

11.34 Proximal Methods

We will look at the composite function:

$$f(x) = g(x) + h(x)$$

where $g(x)$ is convex and differentiable, while $h(x)$ is convex, but potentially non-differentiable.
Example - Constrained Optimization:

$$\min_{x\in\Omega} \quad f(x) = \min_{x\in\mathbb{R}^d} \quad f(x) + I_\Omega(x)$$

where,

$$I_\Omega = \begin{cases} 0, \text{ if } x \in \Omega \\ \infty, \text{ otherwise} \end{cases}$$

Proximal method generalizes the Projected Gradient Descent.
Compressed Sensing: For, $A \in \mathbb{R}^{n\times d}$, and $b \in \mathbb{R}^n$ we have,

$$\begin{aligned} &\min_{x,s.t.\|x\|_1 \leqslant R} \quad \|Ax-b\|^2 \\ \implies &\min_{x\in\mathbb{R}^d} \quad (\|Ax-b\|^2 \quad + \quad \lambda \|x\|_1) \end{aligned}$$

Proximal Operator: We define, $prox_h : \mathbb{R}^d \to \mathbb{R}^d$

$$prox_h(x) = \arg\min_{y\in\mathbb{R}^d} (h(y) + \frac{1}{2}\|y-x\|^2)$$

where h is the function with which *prox* is to be computed.

11.35 Example (Projection)

Example 11.35.1. *Let $h(x) = I_\Omega(x)$. Then we have*

$$\begin{aligned} prox_h(x) &= \arg\min_{y\in\mathbb{R}^d} (I_\Omega(y) + \frac{1}{2}\|y-x\|^2) \\ &= \arg\min_{y\in\Omega} (\frac{1}{2}\|y-x\|^2) \\ &= P_\Omega(x) \end{aligned}$$

Lecture 12 — September 12

Lecturer: Avishek Ghosh *Scribe: Bhavya Kohli*

12.36 Recap

To recap our motivations, and the proximal operator defined in the previous lecture:

- Our current setup/aim: to minimize objectives which can be expressed as a sum of a "nice" function $f(x)$ (convex, differentiable, maybe with additional structure such as smoothness or strong convexity) and a convex **but possibly non-differentiable** function $h(x)$. i.e.

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

- Our motivation/applications of this: Compressed sensing (use of LASSO operator (for information on this operator check https://www.publichealth.columbia.edu/research/population-health-methods/least-absolute-shrinkage-and-selection-operator-lassothis out)
- The proximal operator $\text{prox}_h(x)$: we defined the operator as follows:

$$\text{prox}_h(x) = \arg\min_{y\in\mathbb{R}^d}\{h(y)+\frac{1}{2}\|y-x\|^2\}$$

- An example of using the proximal operator for the indicator example:

$$\begin{aligned}\text{prox}_{\mathrm{I}_\Omega}(x) &= \arg\min_{y\in\mathbb{R}^d}\{\mathrm{I}_\Omega(y)+\frac{1}{2}\|y-x\|^2\}\\ &= \arg\min_{y\in\Omega}\{\frac{1}{2}\|y-x\|^2\}\\ &= \mathbb{P}_\Omega(x)\end{aligned}$$

  Recall that we could use the definition of the indicator function (being infinite outside of Ω to our advantage and claim that the minimizer would lie in Ω itself, leading to the equivalence of prox to the euclidean projection in this case.

12.37 Proximal operator on the L1-norm

Objective: $h(x) = \lambda\|x\|_1$

$$\begin{aligned}
\text{prox}_h(x) &= \arg\min_{y\in\mathbb{R}^d}\{\lambda \|x\|_1 + \frac{1}{2}\|y-x\|^2\} \\
&= \arg\min_{y\in\mathbb{R}^d}\{\lambda \sum_{i=1}^{d}|x_i| + \frac{1}{2}\sum_{i=1}^{d}(y_i-x_i)^2\} \\
&= \arg\min_{y\in\mathbb{R}^d}\{\sum_{i=1}^{d}(\lambda|x_i| + \frac{1}{2}(y_i-x_i)^2)\} \\
&= \arg\min_{y\in\mathbb{R}^d}\{\sum_{i=1}^{d}\phi_i(y_i)\}
\end{aligned}$$

In this case, using the above decomposition, we can solve d individual 1-d problems instead of dealing with a single d-dimensional problem. Note that since $|y_i|$ is non-differentiable, we work with the sub-differential of $\phi_i(y_i)$. i.e.

$$\delta\phi_i(y_i) = \{\lambda z_i - (x_i - y_i)\}$$

Where $z_i = \text{sign}(z_i)$ if $z_i \neq 0$, and $z_i \in [0,1]$ otherwise. Also note that in any case, $|z_i| \leqslant 1$

From the first order optimality condition, for y_i^* to be the solution,

$$0 \in \delta\phi_i(y_i^*)$$

Rearranging the terms, and using the strong convexity property of $\phi_i(y_i)$ (because of the strongly convex $(y_i - x_i)^2$ term) for the equality condition,

$$y_i^* = x_i - \lambda z_i$$

We now consider 3 cases to estimate how y_i^* varies with different values of x_i:

Case 1: if $x_i > \lambda$

$$\begin{gathered}
y_i^* = x_i - \lambda z_i > 0 \\
\text{sign}(y_i^*) = 1 \\
\implies z_i = 1 \\
\therefore \mathbf{y_i} = \mathbf{x_i} - \lambda
\end{gathered}$$

Case 2: if $x_i < -\lambda$

$$\begin{gathered}
y_i^* = x_i - \lambda z_i < 0 \\
\text{sign}(y_i^*) = -1 \\
\implies z_i = -1 \\
\therefore \mathbf{y_i} = \mathbf{x_i} + \lambda
\end{gathered}$$

Case 3: if $y_i = 0$

$$x_i = \lambda z_i$$
$$\therefore \mathbf{x_i} \in [-\lambda, \lambda]$$

Combining:

$$y_i(x_i) = \begin{cases} x_i + \lambda & \text{if } x_i < -\lambda \\ 0 & \text{if } x_i \in [-\lambda, \lambda] \\ x_i - \lambda & \text{if } x_i > \lambda \end{cases}$$

This result is also known as the **Soft Thresholding Function** (https://dsp.stackexchange.com/questions/24156/und soft-thresholding-operatorthis is a decent explanation of the motivation, and the linked document goes in-depth into the derivation and deep analysis, for those interested.)

Applying this over d-dimensions, we get the final proximal version of $h(x) = \lambda \|x\|_1$ as follows:

$$\text{prox}_h(x) = \mathbb{S}_\lambda(x)$$

Where $\mathbb{S}_\lambda$ is the soft thresholding function applied on each dimension of x.

12.37.1 Proximal operator on the gradient descent update

For a differentiable objective $f : \mathbb{R}^d \to \mathbb{R}$, define a new function $h(x)$ as the linearization of $f(x)$ around a fixed point y, and scale it by α:

$$\alpha h(x) = \alpha\{f(x) + \langle \nabla f(x), y - x \rangle\}$$

The prox operator applied to this function would be:

$$\text{prox}_{\alpha h}(x) = \underset{y \in \mathbb{R}^d}{\arg\min}\{\alpha f(x) + \alpha \langle \nabla f(x), y - x \rangle + \frac{1}{2} \|y - x\|^2\}$$

Differentiating the argmin argument w.r.t y,

$$\alpha \nabla f(x) + y - x$$

Setting this to zero, the optimum y is therefore given by

$$y^* = x - \alpha \nabla f(x)$$

which is our familiar gradient descent update step.

Another way to look at it

1. At iteration k, we have x^k, objective f
2. Assume f is linear in (small) neighbourhood of x^k
3. Hence try to "optimize" by using the linearized version
4. Add a quadratic regulizer term so as to not go too far from the point (which will cause greater errors due to the linearization assumption)
5. The optimal next point in this case is **equivalent** to the gradient descent update

12.38 Proximal algorithms

12.38.1 Pure-proximal

- Suppose the objective f is convex
- At iterate k, compute and set $\text{prox}_{\alpha f}(x^k) = x^{k+1}$, where $\alpha \geqslant 0$ is the step size

In the above setting:

$$x^{k+1} = \underset{y\in\mathbb{R}^d}{\arg\min}\{\alpha f(y) + \frac{1}{2}\|y-x\|^2\} = \underset{y\in\mathbb{R}^d}{\arg\min}\{f(y) + \frac{1}{2\alpha}\|y-x\|^2\}$$

Remark 12.51. *Despite how circular this argument appears (the fact that we are using f in an optimization problem in order to get iterates in order to optimize f), we will see examples where this is actually not a bad idea.*

Example 12.38.1. *structure of f matters:*

- *f is smooth and convex: in this case, the argmin objective $g(y) = f(y) + \frac{1}{2\alpha}\|y-x\|^2$ is smooth and **strongly convex**. As seen in the algorithm demo in one of the earlier classes, compared to convex objectives, the minimization of strongly convex objectives is extremely fast, and hardly takes 3-5 epochs.*
- *f is smooth and convex, but with high condition number: in this case, adding the $\frac{1}{2\alpha}\|y-x\|^2$ term **reduces the condition number**, and provides the same benefit as the case above (fast convergence for computing iterate).*

12.38.2 Fixed point of this Pure proximal algorithm

We can also ask, if x^* is the optimum, does the following hold true:

$$x^* = \text{prox}_{\alpha f}(x^*)$$

Alternatively, to prove correctness, we need to show that

$$x^* = \underset{y\in\mathbb{R}^d}{\arg\min}\{f(y) + \frac{1}{2\alpha}\|y-x^*\|^2\} = \underset{y\in\mathbb{R}^d\{g(y)\}}{\arg\min}$$

From the first order optimality condition, $0 \in \delta g(y^*)$, i.e.,

$$0 \in \{\delta f(y^*) + \frac{1}{\alpha}(y^* - x^*)\}$$

If x^* is a fixed point, the result of the argmin would also be x^*, i.e., $y^* = x^*$. Plugging this in the above condition, we have

$$0 \in \delta f(x^*)$$

Which is true since x^* is the optimium of f.

An interesting interpretation

Relating optimization to a set of differential equations..

- For continuous t: $\frac{dx(x)}{dt} = -\nabla f(x(t))$. This can be thought of as a system of d ODE's
- At steady state: $\frac{dx(t^*)}{dt} = 0 = \nabla f(x^*)$. This is the standard FONC, and will imply $x(t^*) = x^* =$ optimum
- Discretization (for small positive α):
  1. Forward-Euler:
  $$\frac{dx(t)}{dt}|_x^k \approx \frac{x^{k+1} - x^k}{\alpha} = -\nabla f(x^k)$$
  $$\implies x^{k+1} = x^k - \alpha \nabla f(x^k)$$

  Forward-Euler relates to Gradient Descent

  2. Backward-Euler:
  $$\frac{dx(t)}{dt}|_x^{k+1} \approx \frac{x^{k+1} - x^k}{\alpha} = -\nabla f(x^{k+1})$$
  $$\implies x^{k+1} + \alpha \nabla f(x^{k+1}) = x^k$$

  Instead of a statement saying "x^{k+1} is equal to ...", we now have "x^{k+1} satisfies ...".
  Connection to pure-prox: we have the output of the armin, $y^* = x^{k+1}$. Computing grad w.r.t. y at this optimum, we get,
  $$\alpha \nabla f(x^{k+1}) + (x^{k+1} - x^k) = 0$$

  Backward-Euler relates to pure-prox

Last Class:

- Proximal gradient methods.
- Convergence guarantee for m-**strongly convex** and M-**smooth** differentiable function part.

Today:

- Convergence guarantee for **convex** and M-**smooth** differentiable function part.
- Proximal decomposition.
- Stochastic optimization.

14.39 Recap from the last class

In the last class we saw proximal gradient methods, in particular,the convergence guarantee of the proximal gradient method when the differentiable part of the function has nice structures (m-**strongly convex** and M-**smooth**). We are interested in solving the following optimization problem:

$$\min_{x\in\mathbb{R}^d} \quad f(x)+h(x). \tag{74}$$

$f(x)$ is the convex and differentiable part of the objective function and $h(x)$ is the convex and potentially non-differentiable part of the objective function. The convergence guarantee of proximal gradient methods with m-**strongly convex** and M-**smooth** $f(x)$, and with the step size $\alpha^k = \frac{2}{M+m}$ is as follows:

$$\left\|x^k - x^*\right\| \leqslant \left(\frac{1-\frac{m}{M}}{1+\frac{m}{M}}\right)^k \left\|x^0 - x^*\right\|. \tag{75}$$

We proved the guarantee (75) using the **non-expansive property** of the **prox** operator, which we discussed in the last lecture, Lecture 13.

14.40 Another Guarantee of Proximal Gradient Methods

We will relax the structure of the differentiable part of the objective function, i.e. $f(x)$, from m-**strong convex** and M-**smooth** to **convex** and M-**smooth** only. We will see and prove the guarantee of proximal gradient methods.

Suppose $\phi : \mathbb{R}^d \longrightarrow \mathbb{R}$ be a convex function such that it can be written as a sum of the differentiable convex function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ and potentially non-differentiable convex function $h : \mathbb{R}^d \longrightarrow \mathbb{R}$. Our optimization problem then becomes:

$$\min_{x\in\mathbb{R}^d} \quad \phi(x) = \min_{x\in\mathbb{R}^d} \quad f(x)+h(x). \tag{76}$$

Theorem 14.52. *Suppose ϕ is convex function and is the sum of convex and M-smooth and only convex functions. Our optimization problem is* (76)*. We run Proximal Gradient with $\alpha^k = \frac{1}{M}$ for T iterations. We obtain*

$$\phi(x^T) - \phi(x^*) \leqslant \frac{M}{2T} \left\|x^0 - x^*\right\|^2, \tag{77}$$

where $\phi(x) \coloneqq f(x) + h(x)$ and x^* is the minimizer of the function $\phi(x)$, the differentiable function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ is assumed to be M-smooth and convex, and the potentially non-differentiable function $h : \mathbb{R}^d \longrightarrow \mathbb{R}$ is convex.

Proof: The proof for this guarantee is similar to the proof of PGD guarantee for M-smooth and convex functions. Denote the update,

$$T(x) = \text{prox}_{h,M}\left[x - \frac{1}{M}\nabla f(x)\right]. \tag{78}$$

Similar to the PGD proof, we define **"Gradient-Type"** operator, which is also known as the residual.

$$g(x) = M(x - T(x)). \tag{79}$$

$g(x)$ plays the role of the so-called pseudo gradient. We can write,

$$T(x) = x - \frac{1}{M}g(x). \tag{80}$$

Iteratively, the updates of Proximal Gradient satisfy,

$$\begin{aligned} x^{k+1} &= T(x^k) \\ x^{k+1} &= x^k - \frac{1}{M}g(x^k). \end{aligned} \tag{81}$$

We should be able to show **"Descent Lemma"** under same pseudo gradient just like the case of PGD.

Lemma 14.53. ***(Descent Lemma)*** *For all $x, y \in \mathbb{R}^d$,*

$$\phi(T(x)) - \phi(y) \leqslant \langle g(x), x - y\rangle - \frac{1}{2M}\|g(x)\|^2 \tag{82}$$

To observe the descent behaviour, let $x = y$,

$$\begin{aligned} \phi(T(x)) - \phi(x) &\leqslant \langle g(x), x - x\rangle - \frac{1}{2M}\|g(x)\|^2 \\ \phi(T(x)) - \phi(x) &\leqslant -\frac{1}{2M}\|g(x)\|^2 \end{aligned} \tag{83}$$

We have 1 step progress in function value and clearly descent behaviour is there in (83) due to negative upper bound. We already proved the Descent Lemma for the case of PGD and it goes same here as well, hence it should be same, just need to use the consistent notations.

For a proof of the Descent Lemma in 14.53 and similarly the Theorem in 14.52, we refer the readers to [9, Chapter 9 -: Lemma 9.4 and Theorem 9.5]. □

14.41 Proximal (Moreau) Decomposition

Definition 14.54. *The **(Convex Conjugate)** of a given closed function f is defined by*

$$f^*(x) = \sup_y \ [\langle x, y\rangle - f(y)]. \tag{84}$$

The convex conjugate is also known as Fenchel Dual or Convex Dual. We can decompose any $x \in \mathbb{R}^d$ using proximal decomposition as

$$x = \text{prox}_f(x) + \text{prox}_{f^*}(x). \tag{85}$$

We can show that the convex conjugate of convex f, i.e. f^* is also convex, because it is nothing but the point-wise supremum of a concave function. The expression whose point-wise supremum we are finding is concave due linear function being concave and negative of the convex function is concave, hence sum of two concave functions is concave and it's point-wise supremum is convex, which is the pretty standard result.

The decomposition in (85) is also known as Moreau Decomposition and we will not prove it. Still, it depends on first order condition for prox operator. For a proof of the Moreau Decomposition in (85), we refer the readers to [7, Section 2.5].

14.41.1 Examples

- **Vector Decomposition**
  Consider a linear subspace $L \subseteq \mathbb{R}^d$. A linear subspace is a vector space that is closed under addition and scalar multiplication. Let us define the indicator function

$$\begin{aligned} f(x) &= \mathbf{1}_L(x) \\ \mathbf{1}_L(x) &:= \begin{cases} 0 & \text{if } \ x \in L\,, \\ \infty & \text{if } \ x \notin L\,. \end{cases} \end{aligned} \tag{86}$$

  Define the orthogonal complement of L, denoted by $L^\perp$ as

$$L^\perp = \{y : \langle x, y\rangle = 0, \forall x \in L\}. \tag{87}$$

  Using (84) we can find the convex conjugate function of f as:

$$f^*(x) = \sup_y \ [\langle x, y\rangle - f(y)]. \tag{88}$$

  Then, $\forall x \in L^\perp$ we have $f^*(x) = 0$, as $\langle x, y\rangle = 0$, which follows from the definition of the $L^\perp$. And $\forall x \notin L^\perp$ we have $f^*(x) = \infty$, as some component of x is aligned with L and hence we can scale it as large as possible. By observing the function behaviour of convex conjugate of f, we can write

$$\begin{aligned} f^*(x) &= \mathbf{1}_{L^\perp}(x) \\ \mathbf{1}_{L^\perp}(x) &:= \begin{cases} 0 & \text{if } \ x \in L^\perp\,, \\ \infty & \text{if } \ x \notin L^\perp\,. \end{cases} \end{aligned} \tag{89}$$

Now with $f(x) = \mathbf{1}_L(x),\ f^*(x) = \mathbf{1}_{L^\perp}(x)$, we can argue via linear subspace argument that any vector can be projected to any linear subspace and its orthogonal complement, sum of these projections is that vector itself.

$$\begin{aligned} \text{prox}_f(x) &= \text{prox}_{\mathbf{1}_L}(x) = P_L(x) \\ \text{prox}_{f^*}(x) &= \text{prox}_{\mathbf{1}_{L^\perp}}(x) = P_{L^\perp}(x). \end{aligned} \tag{90}$$

The prox operator with respect to the function f is defined as

$$\text{prox}_f(x) = \underset{y\in\mathbb{R}^d}{\arg\min}\{f(y) + \frac{1}{2}\|y-x\|^2\}. \tag{91}$$

From (91), we see that the prox operator is a mapping from $\mathbb{R}^d$ to $\mathbb{R}^d$. Now we can evaluate this prox operator for f which is an indicator function as defined in (86) and hence

$$\begin{aligned} \text{prox}_f(x) &= \underset{y\in\mathbb{R}^d}{\arg\min}\{f(y) + \frac{1}{2}\|y-x\|^2\} \\ \text{prox}_f(x) &= \underset{y\in\mathbb{R}^d}{\arg\min}\{\mathbf{1}_L(y) + \frac{1}{2}\|y-x\|^2\} = \underset{y\in L}{\arg\min}\{\frac{1}{2}\|y-x\|^2\} = P_L(x), \end{aligned} \tag{92}$$

where $P_L(x)$ is the standard Euclidean projection of the vector x onto the linear subspace L. On a similar note we see that

$$\begin{aligned} &\text{prox}_f(x) = P_L(x) \\ &\text{prox}_{f^*}(x) = P_{L^\perp}(x) \\ &P_L(x) = \underset{y\in L}{\arg\min}\{\frac{1}{2}\|y-x\|^2\} \\ &P_{L^\perp}(x) = \underset{y\in L^\perp}{\arg\min}\{\frac{1}{2}\|y-x\|^2\}. \end{aligned} \tag{93}$$

Based on our projection argument of any vector x on any linear subspace L and its orthogonal complement $L^\perp$, we have

$$x = P_L(x) + P_{L^\perp}(x) \implies x = \text{prox}_f(x) + \text{prox}_{f^*}(x). \tag{94}$$

Hence we showed the proximal decomposition for the case of vector decomposition. This is already familiar result to us as it follows directly from fundamentals of the linear algebra.

- **Cone Decomposition**
  Let $\mathscr{K}$ be the set which represents any cone, then we can define the polar cone as

$$\mathscr{K}^* = \{y : \langle x, y\rangle \leqslant 0\ \forall x \in \mathscr{K}\}. \tag{95}$$

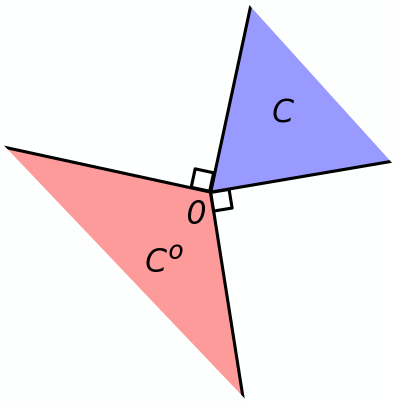


Figure 14.11. Cone(C or $\mathscr{K}$) and the Polar Cone(C^0 or $\mathscr{K}^*$) or the vice-versa.

Similar to the vector decomposition example, here we can also define the indicator function for the cone and the polar cone as

$$f(x) = \mathbf{1}_{\mathscr{K}}(x),\ f^*(x) = \mathbf{1}_{\mathscr{K}^*}(x). \tag{96}$$

By following the same method as in vector decomposition, we may show easily that

$$x = P_{\mathscr{K}}(x) + P_{\mathscr{K}^*}(x) \implies x = \text{prox}_f(x) + \text{prox}_{f^*}(x). \tag{97}$$

Hence we showed the proximal decomposition for the case of cone decomposition as well. This is somewhat unfamiliar result to us unlike the vector decomposition example.

14.42 Stochastic Optimization

Up until now we have deterministic scenarios and the algorithms, and we have the following flow:

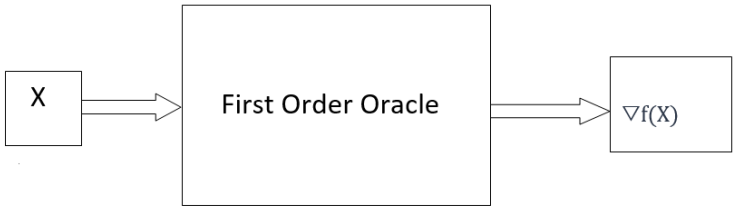


Figure 14.12. Deterministic first order oracles.

We can also have a scenario which is deterministic with some imperfection which is due to fixed disturbances. And the flow of it is as follows:

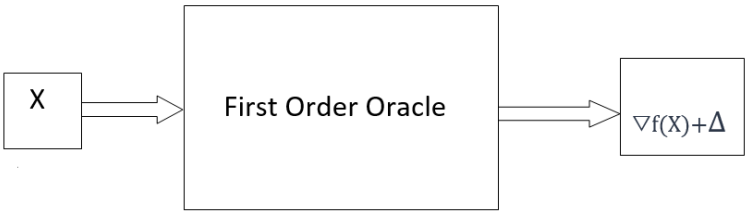


Figure 14.13. Deterministic and imperfect first order oracles.

where X is the query point, $\nabla f(X)$ is the output and Δ is the fixed disturbances.

14.42.1 Stochastic System

In the setting of the stochastic system, we have

$$(x, \xi) \longrightarrow G(x; \xi). \tag{98}$$

where x is the input, ξ is the randomness and $G(x;\xi) \in \mathbb{R}^d$ is the random vector which is kind of proxy for the gradient. We typically ensure the property of unbiasedness which is

$$\mathbb{E}_\xi[G(x;\xi)] = \nabla f(x). \tag{99}$$

14.43 Stochastic Gradient Method

We are at x^k, the next iterate using normal GD update with gradient replaced by its proxy which satisfies (99), can be written as

$$x^{k+1} = x^k - \alpha^k G(x^k; \xi^k), \tag{100}$$

where $G(x^k;\xi^k)$ can be termed as Stochastic Gradient (SG) and whichever algorithm uses this update is known as Stochastic Gradient Method (SGM).

Remark 14.55. *"On average" we are making gradient-steps, because of the unbiasedness property given in* (99).

14.43.1 Examples

- **Large Scale Machine Learning (ML) Problems**
  We may have a loss function which is the average of the loss functions for individual data points and it can be written as

  $$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x). \tag{101}$$

  This is also called the finite sum problem. n can be the number of data points, f_i is the fit/loss on the i^{th} data point. Typically n is large and we have $\nabla f(x) \approx O(n)$ computations.

– **Case-1 (Regression Examples)**
We can define the individual functions in finite sum of (101) as

$$f_i(x) = \frac{1}{2}(b_i - \langle a_i, x\rangle)^2, \tag{102}$$

where $f_i(x)$ is known as squared error loss, b_i is the response and a_i is the covariates.

– **Case-2 (Classification Examples)**
Again we can define the individual functions in finite sum of (101) as

$$f_i(x) = \log_e(1 + e^{b_i\langle a_i, x\rangle}), \tag{103}$$

where $f_i(x)$ is known as logistic loss, b_i is the response and a_i is the covariates.
In case of the binary or the two class classification, we have $b_i \in \{-1, +1\}$ and $a_i \in \mathbb{R}^d$.

14.43.2 Randomized Approximation

Consider a finite sum problem of (101), now we will use randomization concept in a sense that our randomness would be over the index of the data point and hence we have $\xi \in \{1, 2, ..., n\}$ uniformly at random (u.a.r.), i.e. $P(\xi = j) = \frac{1}{n}$, $\forall j = 1, 2, ..., n$. We define

$$G(x; \xi) = \nabla f_\xi(x). \tag{104}$$

Taking the expectation of the gradient proxy over the randomness,

$$\begin{aligned} \mathbb{E}_\xi[G(x;\xi)] &= \sum_{j=1}^{n} \nabla f_{\xi=j}(x) P(\xi = j) \\ \mathbb{E}_\xi[G(x;\xi)] &= \sum_{j=1}^{n} \nabla f_{\xi=j}(x) \frac{1}{n} = \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(x). \end{aligned} \tag{105}$$

14.43.3 Observations

- We are gaining in terms of computation, i.e. we are saving a lot in terms of the computations $\approx O(n)$.
- We are losing in terms of the gradient quality as it goes down (we will make this more formal in the next class).
- We have a parallel architecture, aggregator and hence the algorithm can be parallelized, which happened most recent and goes by the name "Federated Learning System".

15.44 Last Lecture

Class of problems we were dealing with: unconstrained, stochastic optimization problems with the following data:

- We have input of the oracle as x and output is $G(x;\xi)$
- where G is Stochastic gradient
- x is query point
- and ξ is a random variable

Stochastic Gradient Descent:

$$x^{k+1} = x^k - \alpha^k \cdot G(x;\xi)$$

Unbiased Assumption:

$$\mathbb{E}_\xi[G(x;\xi)] = \nabla f(x)$$

Boundedness Assumption: We assume a structure on the second moment,

$$\mathbb{E}_\xi[\|G(x;\xi)\|^2] \leqslant B^2.$$

Let us see some examples:

Example 15.44.1. *Suppose we have a stochastic gradient $G(x;\xi)$ represented as the overall gradient $\nabla f(x)$ and noise $W(\xi)$ i.e.*

$$G(x;\xi) = \nabla f(x) + W(\xi)$$

Unbiasedness: *It follows the unbiasedness assumption i.e.*

$$\mathbb{E}_\xi[W(\xi)] = 0$$

Boundedness: *Suppose f is Lipschitz, i.e., $\|\nabla f(x)\| \leqslant L$ for any x. With this, we have*

$$\mathbb{E}[\|G(x;\xi)\|^2] = L^2 + \mathbb{E}[\|W(\xi)\|^2] \leqslant B^2$$

where, $L^2 + \mathbb{E}[\|W(\xi)\|^2] = B^2$.

Example 15.44.2. ***(Finite Sum Problem)*** *Consider the function f as sum of n individual functions $f_i(x)$, which is defined by*

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

We consider $\xi \in \{1,2,3,\ldots,n\}$ u.a.r. and compute $G(x;\xi) = \nabla f_\xi(x)$, where G is a stochastic gradient. Also, $W(\xi) = \nabla f_\xi(x) - \nabla f(x)$ where, $\nabla f_\xi(x) = G(x;\xi)$ and $\mathbb{E}[W(\xi)] = 0$.

Example 15.44.3. ***(Stochastic coordinate model)*** *We define d-dimensional function $f(x)$*

$$f(x) = f(x_1, x_2, \ldots, x_d).$$

Consider $\xi \in \{1,2,3,\ldots,n\}$ chosen u.a.r. and $P(\xi = i) = \frac{1}{d}$ for all $i = 1,2,\ldots,d$. We compute

$$G(x;\xi) = \left[\frac{\partial f(x)}{\partial x_\xi}\right] e_\xi \text{ where } \mathbf{e}_\xi^\intercal = [0 \ldots 1 \ldots 0].$$

Now from unbiasedness

$$G(x;\xi) = \left[d\left(\frac{\partial f(x)}{\partial x_\xi}\right)\right] e_\xi$$

$$\mathbb{E}[G(x;\xi)] = \sum_{i=1}^{d}\left(\frac{1}{d}\right)\left[d \cdot \frac{\partial f(x)}{\partial x_i}\right] e_i = \nabla f(x).$$

Looking at the second moment it is clear that:

$$\mathbb{E}_\xi \|G(x;\xi)\|^2 \leqslant dL^2 \Rightarrow \mathbb{E}_\xi \|G(x;\xi)\|^2 \leqslant d\|\nabla f(x)\|^2, \text{ if } f \text{ is } L\text{-Lipschitz.}$$

Remark 15.56. *The following observations are noteworthy:*

- *The algorithm saves $O(d)$ computation;*
- *Variance of $G(x;\xi)$ is very large.*

For Stochastic Coordinate Descent, we have:

$$\mathbb{E}_\xi \|G(x;\xi)\|^2 \leqslant d\|\nabla f(x)\|^2.$$

Additionally, suppose f is M-smooth, then

$$\mathbb{E}_\xi \|G(x;\xi)\|^2 \leqslant d\|\nabla f(x) - \nabla f(x^*)\|^2 \Rightarrow \mathbb{E}_\xi \|G(x;\xi)\|^2 \leqslant dM\|x - x^*\|^2$$

Theorem 15.57. *Let f be a convex function. We observe stochastic (sub)gradients that are:*

(i) Unbiased

(ii) Satisfy Boundedness

(iii) Independent across rounds

$$G(x^k;\xi^k), \quad \left(\xi^k\right)_{k=1}^T \text{ are independent}$$

Then, for any sequence of positive step sizes,

$$\mathbb{E}\left[f(\bar{x}^T)\right] - f(x^*) \leqslant \frac{\|x^0 - x^*\|^2}{\sum_{k=1}^T \alpha^k} + \frac{B^2 \sum_{k=1}^T (\alpha^k)^2}{\sum_{k=1}^T \alpha^k}$$

where,

$$\bar{x}^T = \frac{\sum_{k=1}^T \alpha^k x^k}{\sum_{k=1}^T \alpha^k}$$

Step-size Selection: R.H.S is exactly the same as sub-gradient method. We choose,

$$\alpha = \frac{R}{B\sqrt{T}}, \quad \|x^0 - x^*\| \leqslant R, \quad \|x_0 - x^*\| \leqslant R, \, T \text{ is the number of times the algorithm is run.}$$

After substitution, we get

$$\mathbb{E}[f(\bar{x}^T)] - \mathbb{E}[f(x^*)] \leqslant \frac{RB}{\sqrt{T}}.$$

Iteration Complexity: $T(\varepsilon) \geqslant \frac{R^2B^2}{\varepsilon^2}$ for ε accuracy. Comparison with Ordinary Sub-gradient Method: Focus on stochastic coordinate descent. Error for the stochastic method

$$T_{\text{stochastic}}(\varepsilon) = \frac{B^2R^2}{\varepsilon^2} \quad \text{or} \quad T_{\text{stochastic}}(\varepsilon) = \frac{dL^2R^2}{\varepsilon^2}.$$

On the other hand, ordinary sub-gradient method,

$$T_{\text{ordinary}}(\varepsilon) = \frac{L^2R^2}{\varepsilon^2}.$$

Remark 15.58. *We record the following important points:*

- *We need d many iterations.*
- *We are not saving in terms of total computation.*

Benefits:

- *Parallelism*
- *Non-uniform sampling*
  - $P(\xi = j) = p_j$
  - *We will choose p_j optimally.*
  - $G(x;\xi) = \frac{1}{p_\xi}\left(\frac{\partial f(x)}{\partial x_\xi}\right) e_\xi$

Proof: (Proof of Theorem 15.57) At round k, we observe

$$G^k = G(x^k; \xi^k)$$

where, x^k is the current iterate and ξ^k is the noise at the current iterate. We define,

$$w^k = G^k - \nabla f(x^k)$$

Unbiasedness:

$$\mathbb{E}_{\xi^k}[w^k] = 0$$

With this,

$$x^{k+1} = x^k - \alpha^k G^k$$

or,

$$x^{k+1} - x^k = -\alpha^k[\nabla f(x^k) + w^k]$$

By convexity,

$$f(x^k) - f(x^*) \leqslant \langle \nabla f(x^k), x^k - x^* \rangle$$

or,

$$f(x^k) - f(x^*) \leqslant \frac{1}{\alpha^k} \left[\langle x^k - x^{k+1}, x^k - x^* \rangle - \langle w^k, x^k - x^* \rangle \right]$$

or,

$$f(x^k) - f(x^*) \leqslant \frac{1}{2\alpha^k} \left(\|x^k - x^*\|^2 + \|x^k - x^{k+1}\|^2 - 2\|x^{k+1} - x^*\|^2 \right) - \frac{1}{\alpha^k} \langle w^k, x^k - x^* \rangle$$

Now,

$$\mathbb{E}_{\xi^k}[f(x^k) - f(x^*)] \leqslant \frac{(\alpha^k)B^2}{2} + \frac{1}{2\alpha^k} \left(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 \right)$$

or,

$$\mathbb{E}_{\xi^k}[\alpha^k f(x^k) - \alpha^k f(x^*)] \leqslant \frac{(\alpha^k)^2 B^2}{2} + \frac{1}{2} \left(\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2 \right)$$

We now consider all randomness up to time k,

$$\mathbb{E}[f(x^k) - f(x^*)] = \mathbb{E}_{\xi_0, \xi_1, \ldots, \xi_{k-1}} \left[\mathbb{E}_{\xi^k}[f(x^k) - f(x^*) | \xi_0, \xi_1, \ldots, \xi_{k-1}] \right]$$

Using the **Law of Iterated Expectation**, we get,

$$\mathbb{E}[\alpha^k f(x^k) - \alpha^k f(x^*)] = (\alpha^k)^2 B^2 + \frac{1}{2} [\mathbb{E}\|x^k - x^*\|^2 - \mathbb{E}\|x^{k+1} - x^*\|^2]$$

Now take the sum $k = 1$ to T and divide both sides by $\sum_{k=1}^{T} \alpha^k$. Hence, we get the final result since f is common. □

16.45 Stochastic Gradient Method (SGM)

$G(x;\xi)$ x = Query Point, ξ = Random Variable Noise
Unbiasedeness $\implies E_\xi G(x;\xi) = \nabla f(x)$ Boundedness (B) $\implies E_\xi \|G(x;\xi)\|^2 \leqslant B^2$
Under these two conditions, SGM converges.

$$E\left[f\left(\bar{x}^T\right)\right] - f\left(x^\star\right) \leqslant O\left(\frac{BR}{\sqrt{T}}\right) \tag{106}$$

$$\text{where } \bar{x} = \frac{\sum_{k=1}^T \alpha^k x^k}{\sum_{k=1} \alpha^k}, \text{R denotes Initialisation}$$

Convergence Rate: $O\left(\frac{1}{\sqrt{T}}\right)$
$\rightarrow$ How can we improve the convergence rate? $\rightarrow$ Impose structure on f :
We will assume that f is $\mu - PL$ (Polyak-Lojasiewicz (PL) condition)

16.45.1 Polyak-Lojasiewicz (PL) Condition

Define: f is $\mu - PL$ if

$$\langle \nabla f(y), y - x^\star \rangle \geqslant \mu \|y - x^\star\|^2 \forall y \tag{107}$$

We will assume that f is $\mu - PL$ (f satisfies PL condition). Thus, we can relax the condition of strong convexity. It can be easily seen that f is u-strongly convex $\Rightarrow \mu - PL$ condition.

16.45.2 Stochastic Gradient Assumption: (A,B) Condition

$$E_\xi\left[\|G(x;\xi)\|^2\right] \leqslant A^2 \|x - x^\star\|^2 + B^2 \tag{108}$$

Example. (Randomised Kaczmarz) Classical numerical optimization.
System of n-linear equations

$$Ax = b \quad \text{where } A = \underbrace{\begin{bmatrix} a_1^\top \\ a_2{}^\top \\ \vdots \\ a_n^\top \end{bmatrix}}_{n \times d \text{ matrix}} \qquad b = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{n \times 1}$$

Solving

$$a_i^\top x = b_i \quad \text{for } i = 1, \ldots, n$$

We could have solved it by calculating A^{-1} or some other standard method but these are infeasible as the n increases.
We can construct a optimization problem.

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(a_i^\top x - b_i\right)^2$$

Suppose System of linear equations admit unique solution $x^\star$

$$\begin{aligned} &Ax^\star = b \\ &a_i^\top x^\star = b_i \text{ for all } i = 1,\ldots,n \\ &f(x) = \frac{1}{2n}\|Ax-b\|^2 \end{aligned}$$

Question: Is f PL?
$\rightarrow$

$$\nabla^2 f(x) = \frac{1}{n}A^\top A \text{ (derived in previous lectures)}$$

Suppose $\lambda_{\min}\left(\frac{1}{n}A^\top A\right) > 0 \implies f$ is strongly convex with $\lambda_{\min} \implies f$ is $\lambda_{\min} - PL$

Check Stochastic Gradient

$\xi \in \{1,\ldots,n\}$ uniformly at random.

$$\begin{aligned} G(x,\xi) &= a_\xi\left(\langle a_\xi, x\rangle - b_\xi\right) \\ E_\xi\left[\|G(x,\xi)\|^2\right] &= \frac{1}{n}\sum_{i=1}^{n}\|a_i(\langle a_i,x\rangle - b_i)\|^2 = \frac{1}{n}\sum_{i=1}^{n}\|a_i\left(\langle a_i,x\rangle - \langle a_i,x^\star\rangle\right\|^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\|a_i\langle a_i, x-x^\star\rangle\|^2 = \frac{g}{2}(x-x^\star)M(x-x^\star) \\ &\text{where } M = \frac{1}{n}\sum_{i=1}^{n}\|a_i\|^2 a_i a_i^\top \\ &\leqslant \lambda_{\max}(M)\|x-x^\star\|^2 \end{aligned}$$

Remark: Satisfies (A,B) condition with $A^2 = \lambda_{\max}(M); B^2 = 0$
Structure: Each f_i is also minimized at $x^\star$

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \quad \text{where} \quad f_i(x) = \frac{1}{2}\left(a_i^\top x - b_i\right)^2$$

If x^* is a solution of $f(x)$ then x^* also solves f_i.

$$E_\xi \|G(x;\xi)\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x)\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x) - \nabla f_i(x^\star)\|_2^2$$
$$f_i \text{ is } M-\text{smooth}$$
$$\leqslant \frac{1}{n}\sum_{i=1}^{n} M^2 \|x-x^\star\|^2 = M^2\|x-x^\star\|^2$$

Remark: This satisfies (AB) condition with $A^2 = M^2, B^2 = 0$.

Theorem 16.59. *Suppose f-satisfies $\mu - PL$ condition and stochastic gradient is (A B) Bounded, Unbiased and Independent across rounds, then we have*

$$\Delta^{k+1} \leqslant \left(1 - 2\alpha^k\mu + \left(\alpha^k\right)^2 A^2\right)\Delta^k + \left(\alpha^k\right)^2 B^2$$
$$\text{where } \Delta^k = E\left[\left\|x^k - x^*\right\|^2\right]$$
$$(1 - 2\alpha^k\mu + \left(\alpha^k\right)^2 A^2) = \text{ contraction factor }, \left(\alpha^k\right)^2 B^2 = \text{ error flow}$$

Remark: Optimizing contracting factor over α^k, we get $\alpha^k = \frac{\mu}{A^2}$
With this α^k, we have

$$\Delta^{k+1} \leqslant \left(1 - \frac{\mu^2}{A^2}\right)\Delta^k + \left(\frac{\mu B}{A^2}\right)^2$$

Un-rolling

$$E\left[\left\|x^k - x^\star\right\|^2\right] \leqslant \left(1 - \frac{\mu^2}{A^2}\right)^k \left\|x^0 - x^\star\right\|^2 + \left(\frac{\mu B}{A^2}\right)^2 \left[1 + \left(1 - \frac{\mu^2}{A^2}\right) + \left(\frac{1-\mu^2}{A^2}\right)^2 + \ldots k \text{ terms}\right]$$
$$\leqslant \left(1 - \frac{\mu^2}{A^2}\right)^K \left\|x^0 - x^\star\right\|^2 + \underbrace{\frac{\left(\frac{\mu B}{A^2}\right)^2}{\left(\frac{\mu^2}{A^2}\right)^2}}_{\text{Total Error Flow}}$$

When is this okay? $\rightarrow$ When precision $\varepsilon >$ Total Error Flow
Remark: When $\beta = 0 \Rightarrow$ Geometric Convergence

Under PL condition $(A,0)$ bound $\Rightarrow$ SGM with constant step size $\Rightarrow \Delta^k \leqslant \left(1 - \frac{\mu^2}{A^2}\right)^k \Delta^0$

Special Case (AB) Bound with $A = 0$

Q) f is $\mu - PL, A = 0, B \neq 0$. Can we obtain an improved rate compared to $O\left(\frac{1}{\sqrt{T}}\right)$?

Ans) $\rightarrow$ Yes

Corollary: $(0,B)$ Bound on $\mu - PL$ Loss

$$E_\xi \|G(x;\xi)\|^2 \leqslant B^2$$

Using Main Theorem 16.1,

$$\Delta^{k+1} \leqslant \left(1-2\alpha^k\mu\right)\Delta^k + \left(\alpha^k\right)^2 B^2$$

Select step-size $\alpha^k = \dfrac{1}{2\mu k} (\to 0 \text{ as } k \to \infty)$

$$\Delta^{k+1} \leqslant \left(1-\frac{1}{k}\right)\Delta^k + \frac{1}{k^2}\frac{B^2}{4\mu^2} \leqslant \left(\frac{k-1}{k}\right)\left\{\frac{k-2}{k-1}\Delta^{k-1} + \frac{1}{(k-1)^2}\frac{B^2}{4\mu^2}\right\} + \frac{1}{k^2}\frac{B^2}{4\mu^2}$$

$$= \left(\frac{k-2}{k}\right)\cdot\Delta^{k-1} + \frac{B^2}{4\mu^2}\frac{1}{k}\left\{\frac{1}{k}+\frac{1}{k-1}\right\}$$

Induction over k,

$$\Delta^{k+1} \leqslant \frac{\Delta^0}{k+1} + \frac{B^2}{4\mu^2}\cdot\frac{1}{(k+1)}\left(\sum_{s=1}^{k+1}\frac{1}{s}\right)$$

$$E\left[\left\|x^{k+1}-x^\star\right\|^2\right] \leqslant \frac{\left\|x^0-x^\star\right\|^2}{k+1} + \frac{B^2}{4\mu^2}\frac{\log(2+k)}{(1+k)}$$

Rate: $O\left(\frac{\log(T)}{T}\right)$ instead of $O\left(\frac{1}{\sqrt{T}}\right)$ as before.
We use the PL-condition crucially.

16.46 Summarize

B-Boundedness $\to O\left(\frac{1}{\sqrt{T}}\right) \quad \left(\alpha = O\left(\frac{1}{\sqrt{T}}\right)\right)$
$\mu - PL$ and (AB) Boundedness

$$\text{If } B = 0 \to \text{ Geometric } e^{-T} \quad (\alpha = \text{ constant })$$

$$\text{If } A = 0, B \neq 0 \to O\left(\frac{\log T}{T}\right) \quad \left(\alpha = O\left(\frac{1}{T}\right)\right)$$

$\longrightarrow$ Step-Size Selection in Stochastic Optimization
Toy Example: $f(x) = \frac{1}{2}x^2 \quad f : R \to R$
f is 1 strongly convex $\Rightarrow 1 - PL$

$$G(x,\xi) \approx f'(x)$$

But, we use $\alpha^k = \frac{c}{k}$ [c $\to$ some step-size parameter]

$$x^{k+1} = x^k - \frac{c}{k}x^k \qquad x^\top = x^0\prod_{k=1}^{T}\left(1-\frac{c}{k}\right)$$

Q) What should be the value of C

Exercise: Show $c \in \left(0, \frac{1}{4}\right)$

$$x^{\top} \geqslant x^0 \cdot \frac{g(c)}{T^c}$$

Suppose is small, $c = 0.01$

$$x^T \geqslant x^0 \cdot \frac{g(0.01)}{T^{0.01}} \Rightarrow O\left(\frac{1}{\varepsilon}\right)^{100} \text{ steps.}$$

Lecture 17 — October 6

Lecturer: Avishek Ghosh *Scribe: Vivek Wadate*

Last class:

In the last class, we discussed about Stochastic Gradient Descent (SGD). We saw what is Polyak-Lojasiewicz (PL) Condition.

PL condition

From the last class, $f: C \subset \mathbb{R}^d \longrightarrow \mathbb{R}$, is μ- PL if it satisfies following condition

$$\langle \nabla f(x), y-x \rangle \big) \geqslant \mu ||y-x||^2 \quad \forall y$$

Remark: If f is m - strongly convex $\implies$ f is m - PL.

Theorem 17.60. *f is μ - PL and G(x:ζ) satisfies (i) Unbiased condition, (ii) (A,B) condition (iii) ε_k to be independent Then,*

$$\nabla^{k+1} \leqslant (1-2\alpha^k\mu + (\alpha^k)^2A^2)\nabla^k + (\alpha^k)^2B^2 \tag{109}$$

Where,

$$\nabla^k = \mathbb{E}||x^k - x^*||^2 \tag{110}$$

Special case,

- *If B=0, Convergence to x^* at an exponential speed*
- *If A=0, Convergence at rate of O $(1/T)$*

Proof: *We have,*

$$\begin{aligned} ||x^{k+1}-x^*||^2 &= ||x^k - \alpha^k G(x^k;\zeta^k) - x^*||^2 \\ &= ||x^k - x^*||^2 - 2\alpha^k \left\langle G(x^k;\zeta^k), x^k - x^* \right\rangle + (\alpha^k)^2 ||G(x^k;\zeta^k)||^2 \end{aligned}$$

Taking expectation w.r.t $\zeta^{k-1}, \zeta^{k-2}....\zeta^0$
Recall that, x^k depends on $\zeta^{k-1}, \zeta^{k-2}....\zeta^0$

$$\begin{aligned} \mathbb{E}[\left\langle G(x^k;\zeta^k), x^k - x^* \right\rangle] &= \mathbb{E}_{\zeta^0,\zeta^1.....\zeta^{k-1}} \mathbb{E}_{\zeta^k} \left\langle G(x^k;\zeta^k) | \zeta^0, \zeta^1, ...\zeta^{k-1}, x^k - x^* \right\rangle \\ &= \mathbb{E}_{\zeta^0...\zeta^k}[\left\langle \nabla f(x^k), x^k - x^* \right\rangle] \\ &= \mathbb{E}[\left\langle \nabla f(x^k), x^k - x^* \right\rangle] \end{aligned}$$

Similarly, We have

$$\mathbb{E}||G(x^k;\zeta^k)||^2 \leqslant A^2\mathbb{E}||x^k - x^*||^2 + B^2$$

Now, Our original equation after taking expectation looks like,

$$\nabla^{k+1} \leqslant (1+(\alpha^k)^2A^2)\nabla^k - 2\alpha^k\mathbb{E}[\left\langle (x^k, x^k - x^* \right\rangle] + (\alpha^k)^2B^2$$

Now from the PL condition,

$$\nabla^{k+1} \leqslant (1 - 2\mu\alpha^k + (\alpha^k)^2A^2) + (\alpha^k)^2B^2$$

□

17.47 Problem of Stochastic Gradient Descent

SGD is much faster but the convergence path of SGD is noisier than that of original gradient descent. This is because in each step it is not calculating the actual gradient but an approximation.

17.47.1 Variance Reduction

Naive : - Mini-batch SGD.
instead of quarrying to one point, Query a batch of size r.

$$G(x) = \frac{1}{r}\Sigma_{j=1}^{r} G(x;\zeta^j) \tag{111}$$

Where, ζ^j N {1,2,...n}

- Var (G(x)) gets reduced by factor of r
- Query r times (Computationally expensive by factor of r)

Remark 17.61. *Usually r is taken as a hyper-parameter.*

17.48 Stochastic variance-reduced gradient (SVRG)

Finite sum problem $\implies$ $f(x) = \frac{1}{n}\Sigma_{i=r}^{n} f(x)$

Use current estimate $\bar{x}$ of x^k.

$$V(x^k;\bar{x},\zeta^k) = \nabla f_\zeta(x^k) - \nabla f_\zeta(\bar{x}) + \nabla f(\bar{x}) \tag{112}$$

- Note that we require full gradient at $\bar{x}$.
- We don't want to compute $\bar{x}$ often.
- We fix $\bar{x}$. Run several iterations of SGD.

Fix current guess, $\bar{x}$ run N iterations of SGD.

$$x^{k+1} = x^k - \alpha^k V(x^k;\bar{x},\zeta^k)$$

SVRG is epoch based algorithm.

17.48.1 Unbiasedness

$$\mathbb{E}V(x^k,\bar{x},\zeta) = \frac{1}{n}\Sigma_{i=1}^{n}\nabla f_i(x^k) - \frac{1}{n}\Sigma_{i=1}^{n}\nabla f_i(\bar{x}) + \frac{1}{n}\Sigma_{i=1}^{n}\nabla f_i(\bar{x})$$

$$\mathbb{E}V(x^k,\bar{x},\zeta) = \nabla f(x^k)$$

Hence SVRG is unbiased.

Lemma 17.62. *Suppose f is M-smooth. We have,*

$$\mathbb{E}||V(x^k;\bar{x},\zeta^k)||^2 \leqslant 4M[(f(x^k) - f(x^*) + f(\bar{x}) - f(x^*)]$$

When k is large,

$$f(x^k) \implies f(x^*)$$

$$f(x^k) \implies f(\bar{x})$$

Therefore,
$\mathbb{E}||V(x^k;\bar{x},\zeta^k)||^2$ *Becomes small as k increases.*

Remark 17.63. *This lemma implies (A,B) bound with B=0.*

Theorem 17.64. *If f is m-strongly convex, M-smooth. After N iterations,* $\alpha^k = \frac{1}{8M}$*,*
We have,

$$\mathbb{E}[f(x)] - f(x^*) \leqslant (\frac{32}{3N}.\frac{M}{m} + \frac{1}{3})[f(\bar{x}) - f(x^*)] \tag{113}$$

Where, $x^N = \frac{1}{N}\Sigma_{k=1}^{N}x^k$

Choose, N=64 $\frac{M}{m}$

Lecture 18 — October 13

Lecturer: Avishek Ghosh *Scribe: Reema Deori*

Recap:

1. **GD Algorithm:** We studied the gradient descent algorithm with the iteration at time $t = k$ given by
$$x^{k+1} = x^k - \alpha^k \nabla f(x^k).$$
Further, we stipulate the following assumptions:
   - f is strongly convex and smooth: Rate $\approx e^{-T}$
   - f is convex and smooth: Rate $\approx \frac{1}{T}$
   - f is smooth: Rate $\approx \frac{1}{\sqrt{T}}$
2. **Variations of GD:** We have also seen several versions of the gradient descent algorithm that are detailed below:
   (a) **Constrained Case:** $\min_{x\in\Omega} f(x)$; PGD; Frank Wolfe
   (b) **Non-smooth:** Involves subgradients; Subgradient descent; $\min_{x\in\Omega} f(x)$; f is non-differentiable
   (c) **Composite function:** $\min_x f(x) + h(x)$; f is differentiable; h is potentially non differentiable; Algos: Pure-proximal and Proximal GD
   (d) **Stochastic Optimization:** Instead of $\nabla f(x)$, we have $G(x;\xi)$; SGD converges for a large class of functions; Variance reduced SGD (SVRG)

Today's agenda is to go beyond GD methods and explore momentum methods.

18.49 Momentum Methods

Upto this point, $\{x^{k+1}\}$ depends only on $\{x^k\}$. We now propose and analyze algorithms where the iterations depend on (x_k, x_{k-1}), i.e.,
$$x^{k+1} = f(x^k, x^{k-1})$$
. We study two popular algorithms:

1. **Algorithm 1: Polyak's Heavy Ball Method**
At time k,
$$x^{k+1} = \underbrace{x^k - \alpha^k \nabla f(x^k)}_{\text{G-step}} + \underbrace{\beta^k (x^k - x^{k-1})}_{\text{Momentum correction}}, \text{ with } x^{-1} = x^0. \tag{114}$$

Remark 18.65. *If $\beta^k = 0$, then this is GD.*

2. **Algorithm 2: Nestrov's Accelerated GD (AGD)**
   At time k,
$$x^{k+1} = \underbrace{x^k - \alpha^k \nabla f(x^k + \beta^k(x^k - x^{k-1}))}_{\text{Gradient on a perturbed point}} + \underbrace{\beta^k(x^k - x^{k-1})}_{\text{Momentum correction}} . \tag{115}$$

Remark 18.66. *Several remarks are in order:*

- *Polyak Heavy Ball Algorithm converges for Quadratic Objectives.*
- *Nesterov AGD converges for strongly convex and smooth (convex and smooth) objectives.*
- *AGD method converges much faster than GD.*
- *Optimality: AGD methods are optimal for a large class of losses.*
- ***Negative Remark for AGD:** Stochastic versions of AGD do not work.*

We will study AGD for the following functions:

- Strongly convex and smooth
- Convex and smooth

18.49.1 Another look at AGD update

We introduce this intermediate sequence $\{y^k\}$.
AGD update:

$$y^k = \underbrace{x^k + \beta^k(x^k - x^{k-1})}_{\text{Momentum step}}$$
$$x^{k+1} = y^k - \alpha^k \nabla f(y^k), \text{ with, } x^{-1} = x^0, y^0 = x^0.$$

Intuition: α^k is small and β^k is large. Thus, the algorithm explores x^k in a better way that GD fails to do so.

Remark 18.67. *β^k can be chosen to be large so that $\{x^k\}$ is "explored" in a better way, when compared with GD.*

18.49.2 Guarantees for m-strongly convex and M-smooth function

Consider the optimization problem

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

Theorem 18.68. *We choose $\alpha^k = \frac{1}{M}, \beta^k = \frac{\sqrt{\frac{M}{m}}-1}{\sqrt{\frac{M}{m}}+1}$. AGD satisfies,*

$$f(x^k) - f(x^*) \leqslant \underbrace{\left(1 - \sqrt{\frac{m}{M}}\right)^k}_{\text{Contraction parameter is different from GD}} \times \underbrace{\{f(x^0) - f(x^*) + \frac{m}{2}||x^0 = x^*||^2\}}_{\text{initial condition}} . \tag{116}$$

Let us compare the above results with what we have already seen:

- **For GD:**
  - $$f(x^k) - f(x^*) \leqslant \left(1 - \frac{m}{M}\right)^k D_0, \tag{117}$$
    where D_0 is the initial condition.
  - If $\mathscr{N}_{GD}(\varepsilon)$ is the oracle complexity, then
    $$\mathscr{N}_{GD}(\varepsilon) \geqslant \frac{M}{m} \log\left(\frac{D_0}{\varepsilon}\right). \tag{118}$$
- **For AGD:** If $\mathscr{N}_{AGD}(\varepsilon)$ is the oracle complexity, then
  $$\mathscr{N}_{AGD}(\varepsilon) \geqslant \sqrt{\frac{M}{m}} \log\left(\frac{D_1}{\varepsilon}\right). \tag{119}$$

Remark 18.69.
- *Number of iterations is much less compared to GD when $\frac{M}{m}$ is large.*
- *When $\frac{M}{m}$ is large and β^k is large, exploration is better.*

18.49.3 Nestrov AGD for convex and M-smooth objective

Theorem 18.70. *With proper choices of β^k, keeping $\alpha^k = \frac{1}{M}$, we obtain*

$$f(x^k) - f(x^*) \leqslant \frac{2M}{(k+1)^2} \underbrace{||x^0 - x^*||^2}_{\text{initial condition}}. \tag{120}$$

- **Compare with GD:**
  - GD: $f(x^k) - f(x^*) \leqslant \frac{2M}{k+1} D_0$
  - Oracle Complexity, $\mathscr{N}_{GD}(\varepsilon) > \frac{2M}{\varepsilon} D_0$
  - Oracle Complexity, $\mathscr{N}_{AGD}(\varepsilon) \geqslant \sqrt{\frac{2M}{\varepsilon} D_1}$

Remark 18.71.
- *Strongly Convex and smooth: AGD is optimal*
  - *No first order algorithm can provide a better oracle complexity (refer to [6] for the proof).*
- *Convex and smooth: AGD is optimal*
  - *No first-order algorithms have a better complexity (will be covered in the next lecture).*

18.49.4 Optimality of GD

Solving $\min_x f(x)$, where f is Lipschitz and convex:

$$||\nabla f(x)|| \leqslant G, \forall x. \tag{121}$$

Theorem 18.72. *GD with $\alpha^k = \frac{c}{\sqrt{K}}$, where c is a constant, satisfies*

$$f\left(\frac{\sum_{k=1}^{T} \alpha^k x^k}{\sum_{k=1}^{T} \alpha^k}\right) - f(x^*) \leqslant \frac{\bar{c}G}{\sqrt{T}} D_0, \tag{122}$$

where D_0 is the initial condition.

Proof: Same as the proof of Sub-gradient descent. □

Remark 18.73. *A Few remarks are in order:*

- *Oracle complexity, $\mathcal{N}_{GD}(\varepsilon) \geqslant \frac{\bar{c}^2 G^2}{\varepsilon^2} D_0^2$.*
- *The catch here is that GD for this class of functions is optimal:*
  - *No first order algorithm gives a better complexity.*
  - *No point in accelerating such functions.*

19.50 Convergence guarantees for Nesterov's AGD

For any vector v^k, $\tilde{v}^k = v^*$, where v^* is the fixed point of $\{v^k\}_{k=1}^{+\infty}$.

19.50.1 m-strongly convex and M-smooth functions

Theorem 19.74. *For the optimization problem* $\min_x f(x)$*, with m-strongly convex and M-smooth f, employing Nesterov's accelerated gradient descent with* $\alpha^k = \frac{1}{M}$ *and* $\beta^k = \frac{\sqrt{\frac{M}{m}}-1}{\sqrt{\frac{M}{m}}+1}$ *at the k-th iteration, and where x^* represents the minimum value, the following inequality holds:*

$$f(x^k) - f(x^*) \leqslant \left(1 - \sqrt{\frac{m}{M}}\right)\left(f(x^0) - f(x^k) + \frac{m}{2}\|x^0 - x^*\|^2\right)$$

Proof: The proof involves constructing the Lyapunov function given below

$$V^k := f(x^k) - f(x^*) + \frac{1}{2}\|\tilde{x}^k - \rho^2\tilde{x}^{k-1}\|^2.$$

Note that here $\frac{1}{2}\|\tilde{x}^k - \rho^2\tilde{x}^{k-1}\|^2$ represents the momentum component of the algorithm with ρ acting like a regularize. Since f is M-smooth and m-strongly convex $\forall\ w, z$

$$f(z) + \langle\nabla f(z), w - z\rangle + \frac{m}{2}\|z - w\|^2 \leqslant f(w) \leqslant f(z) + \langle\nabla f(z), w - z\rangle + \frac{M}{2}\|z - w\|^2. \tag{123}$$

To simplify algebra we also introduce

$$u^k := \frac{1}{M}\nabla f(y^k).$$

Note that y^k converges to $y^* = x^*$ respectively thus, $u^* = 0$ which implies $u^k = \tilde{u}^k$, and

$$V^{k+1} = f(x^{k+1}) - f(x^*) + \frac{M}{2}\|\tilde{x}^{k+1} - \rho^2\tilde{x}^k\|^2,$$

On substituting (123) with $w = x^{k+1}$ and $z = y^k$,

$$\begin{aligned}
V^{k+1} &\leqslant f(y^k) + \langle\nabla f(y^k), x^{k+1} - y^k\rangle + \frac{M}{2}\|x^{k+1} - y^k\|^2 - f(x^*) + \frac{M}{2}\|\tilde{x}^{k+1} - \rho^2\tilde{x}^k\|^2 \\
V^{k+1} &\leqslant f(y^k) - \frac{M}{2}\|\tilde{u}^k\|^2 - f(x^*) + \frac{M}{2}\|\tilde{x}^{k+1} - \rho^2\tilde{x}^k\|^2 \\
V_{k+1} &\leqslant \rho^2(f(y^k) - f(x^*) + M\langle\tilde{u}^k, \tilde{x}^k - \tilde{y}^k\rangle) - \rho^2 M\langle\tilde{u}^k, \tilde{x}^k - \tilde{y}^k\rangle \\
&\quad + (1-\rho^2)(f(y^k) - f(x^*) + M\langle\tilde{u}^k, \tilde{x}^k - \tilde{y}^k\rangle) - (1-\rho^2)M\langle\tilde{u}^k, \tilde{x}^k - \tilde{y}^k\rangle - \\
&\quad \frac{M}{2}\|\tilde{u}^k\|^2 + \frac{M}{2}\|\tilde{x}^{k+1} - \rho^2\tilde{x}^k\|^2.
\end{aligned}$$

Using (123) with $w = x^k$ and $z = y^k$

$$f(y^k) \leqslant f(x^k) - \langle \nabla f(y^k), x^k - y^k \rangle - \frac{m}{2}\|x^k - y^k\|^2.$$

Using (123) with $w = x^*$ and $z = y^k$

$$f(y^k) \leqslant f(x^*) - \langle \nabla f(y^k), x^* - y^k \rangle - \frac{m}{2}\|x^* - y^k\|^2,$$

and substituting the above two equations

$$\begin{aligned} V^{k+1} &\leqslant \rho^2 (f(x^k) - f(x^*) + \frac{M}{2}\|\tilde{x}^k - \rho^2 \tilde{x}^{k-1}\|^2) + R^k \\ R^k &:= \frac{-m\rho^2}{2}\|\tilde{x}^k - \tilde{y}^k\|^2 - \frac{m(1-\rho^2)}{2}\|\tilde{y}^k\|^2 + M\langle \tilde{u}^k, \tilde{y}^k - \rho^2 \tilde{x}^k \rangle - \frac{M}{2}\|\tilde{u}^k\|^2 \\ &\quad + \frac{M}{2}\|\tilde{x}^{k+1} - \rho^2 \tilde{x}^k\|^2 - \frac{\rho^2 M}{2}\|\tilde{x}^k - \rho^2 \tilde{x}^{k-1}\|^2. \end{aligned}$$

Choosing $\rho^2 = \left(1 - \sqrt{\frac{m}{M}}\right)$. Note that $\rho^2 \leqslant 1$. Refer [9] for details

$$R^k = \frac{-M\rho^2}{2}\left(\frac{m}{M} + \sqrt{\frac{m}{M}}\right)\|\tilde{x}^k - \tilde{y}^k\|^2.$$

Substituting this in the expression for V^{k+1}

$$\begin{aligned} V^{k+1} &\leqslant \rho^2 V^k \\ V^k &\leqslant \rho^{2k} V^0 \\ f(x^k) - f(x^*) &= V^k - \frac{1}{2}\|\tilde{x}^k - \rho^2 \tilde{x}^{k-1}\|^2 \\ f(x^k) - f(x^*) &\leqslant V^k \\ f(x^k) - f(x^*) &\leqslant \rho^{2k} V^0 \\ V^0 &= f(x^0) - f(x^*) + \frac{M}{2}\|\tilde{x}^0 - \rho^2 \tilde{x}^0\|^2 \\ V^0 &= f(x^0) - f(x^*) + (1-\rho^2)^2 \frac{M}{2}\|x^0 - x^*\|^2. \end{aligned}$$

Substituting the expression for V^0 in $f(x^k) - f(x^*) \leqslant \rho^{2k} V^0$

$$f(x^k) - f(x^*) \leqslant \left(1 - \sqrt{\frac{m}{M}}\right)^k \left(f(x^0) - f(x^*) + \frac{m}{2}\|x^0 - x^*\|^2\right).$$

This concludes the proof. □

Concatenating dependence on initial conditions $D^0 := f(x^0) - f(x^*) + \frac{m}{2}\|x^0 - x^*\|^2$.

$$f(x^k) - f(x^*) \leqslant \left(1 - \sqrt{\frac{m}{M}}\right)^k D^0 \tag{124}$$

The oracle complexity of an algorithm A to bring about an error less than ε and is denoted by $N_A(\varepsilon)$. From (124) it is clear that the oracle complexity of Nesterov's accelerated gradient descent method is

$$N_{AGD}(\varepsilon) \geqslant \sqrt{\frac{M}{m}} \log\left(\frac{D^0}{\varepsilon}\right).$$

Comparing this to gradient descent with an oracle complexity of

$$N_{GD}(\varepsilon) \geqslant \frac{M}{m} \log\left(\frac{D^0}{\varepsilon}\right).$$

Often optimization problems involve functions with $\frac{M}{m} >> 1$. In these cases, there can be a difference of orders of magnitude between $\sqrt{\frac{M}{m}}$ and $\frac{M}{m}$. Further, we will also prove that no other first-order algorithm can give a better convergence rate than accelerated gradient descent. This is why Nesterov's accelerated gradient descent is considered to be optimal.

19.50.2 M-smooth and convex functions

Theorem 19.75. *For the optimization problem*

$$\min_x f(x),$$

with convex and M-smooth f, employing Nesterov's accelerated gradient descent with $\alpha^k = \frac{1}{M}$ and an appropriately chosen β^k at the k-th iteration, and where x^* represents the minimum value, the following inequality holds:

$$f(x^k) - f(x^*) \leqslant \frac{2M}{(k+1)^2}\|x^0 - x^*\|^2$$

Proof: Here too we construct a Lyapunov function V^k given by

$$V^k := f(x^k) - f(x^*) + \frac{M}{2}\|\tilde{x}^k - \rho_{k-1}^2 \tilde{x}^{k-1}\|^2.$$

Note that unlike the last proof here ρ is not a constant. Substituting $m = 0$ and $\rho = \rho^k$ in the previous proof we get

$$V^{k+1} \leqslant \rho_k^2 V^k + w^k$$

Here

$$w^k := \frac{M}{2}\|\tilde{y}^k - \rho_k^2 \tilde{x}^k\|^2 - \frac{\rho_k^2 M}{2}\|\tilde{x}^k - \rho_{k-1}^2 \tilde{x}^{k-1}\|^2.$$

We need to choose ρ_k such that $w^k = 0 \; \forall \; k$ This will be true if

$$\tilde{y}^k - \rho_k^2 \tilde{x}^k = \rho_k \tilde{x}^k - \rho_k \rho_{k-1}^2 \tilde{x}^{k-1} \tag{125}$$

From the accelerated gradient descent algorithm, we know that

$$\begin{aligned} y^k &= x^k + \beta^k(x^k - x^{k-1}) \\ y^k &= (1+\beta^k)x^k - \beta^k x^{k-1}, \end{aligned}$$

we know $y^* = x^*$

$$\tilde{y}^k = (1+\beta^k)\tilde{x}^k - \beta^k \tilde{x}^{k-1}.$$

Substituting this in (125)

$$(1+\beta^k)\tilde{x}^k - \beta^k \tilde{x}^{k-1} - \rho^2 \tilde{x}^k = \rho_k \tilde{x}^k - \rho_k \rho_{k-1}^2 \tilde{x}^{k-1}.$$

Equating the coefficients of $\tilde{x}^k$ and $\tilde{x}^{k-1}$

$$\begin{aligned} 1 + \beta^k - \rho_k^2 &= \rho_k \\ -\beta^k &= -\rho_k \rho_{k-1}^2. \end{aligned}$$

Eliminating β^k from these equations gives the following quadratic equation for ρ_k

$$1 + \rho_k(\rho_{k-1} - 1) - \rho_k^2 = 0$$

and solving this for ρ_k

$$\rho_k = \frac{1}{2}(-(\rho_{k-1} - 1) \pm \sqrt{(\rho_{k-1} - 1)^2 + r\rho_k^2})$$

Thus, we have proved for certain values of ρ_k we can enforce the condition $V_{k+1} \leqslant \rho_k^2 V^k$. This proof will continue in the next lecture. □

20.51 Nestorov Accelarated Gradient

Theorem 20.76. *f is convex and M-smooth, we run AGD with $\alpha^k = \frac{1}{M}$ and β^k, chosen approximately.*

$$f(x^k) - f(x^*) \leqslant \frac{2}{(k+1)^2} \left\|x^0 - x^*\right\|^2$$

Proof: We use Lypunov function

$$V_k = f(x^k) - f(x^*) + \frac{M}{2}\|\tilde{x}^k - p_{k-1}^2 \tilde{x}^{k-1}\|^2$$

where

$$\tilde{x}^k = x^k - x^*$$
$$\tilde{x}^{k-1} = x^{k-1} - x^*$$

Goal: Design $\{p_k, \beta^k\}_{k=1}^{\infty}$ such that $w_k = 0$. We have

$$p_k^2 = \frac{(1-p_k^2)^2}{(1-p_{k-1}^2)^2} \quad \text{for every } k = 1, 2, 3, \ldots, n$$

$$V_k \leqslant p_{k-1}^2 V_{k-1} \leqslant p_{k-1}^2 \cdot p_{k-2}^2 \cdots p_1^2 V_1 \leqslant \left(\prod_{j=1}^{k-1} p_j^2\right) V$$

$$\leqslant \frac{(1-p_{k-1}^2)^2}{(1-p_0)^2} V_1$$

We take $p_0 = 0, p_{-1} = 0$. Recall $y^0 = x^0$ from AGD update

$$V_1 \leqslant p_0^2 V_0 + W_0 \leqslant \frac{M}{2} \left\|x^0 - x^*\right\|^2$$

This comes from the definition of W_0.

$$V_1 \leqslant (1-p_{k-1}^2)^2 \frac{M}{2} \left\|x^0 - x^*\right\|^2$$

□

Lemma 20.77. *We have $1 - p_k^2 \leqslant \frac{2}{k+2}$*

Proof: Using Mathematical Induction suppose this holds for k; $1-p_k^2 \leqslant \frac{2}{k+2}$. Want to show $1-P_{k+1}^2 \leqslant \frac{2}{(k+1)+2}$. Show via contradiction suppose convergence holds

$$\begin{aligned} 1-p_{k+1}^2 &> \frac{2}{k+3} \\ p_{k+1}^2 &\leqslant \frac{k+1}{k+3} \\ \frac{(1-p_{k+1})^2}{p_{k+1}^2} &\geqslant \left(\frac{2}{k+3}\right)^2 \frac{k+3}{k+1} = \frac{4}{(k+3)(k+1)} \\ (k+3)(k+1) &\leqslant (k+2)^2 \\ (1-p_k^2)^2 = \frac{(1-p_{k+1}^2)^2}{p_{k+1}^2} &> \frac{4}{(k+2)^2} \\ 1-p_k^2 &\leqslant \frac{2}{k+2} \end{aligned}$$

Contradiction to induction step

$$1-p_{k+1}^2 \leqslant \frac{2}{k+3} \qquad \text{Proved}$$

Combining $V_k \leqslant \frac{2M}{(K+1)^2}\left\|x^0-x^*\right\|^2$ substituting $1-p_{k-1}^2$ using lemma

$$\begin{aligned} f(x^k)-f(x^*) &\leqslant f(x^k)-f(x^*)+\frac{M}{2}\|\tilde{x}^k-p_{k-1}^2\tilde{x}^{k-1}\|^2 = V_k \\ &\leqslant \frac{2M}{(k+1)^2}\left\|x^0-x^*\right\|^2. \end{aligned}$$

□

The following pseudocode gives a breif overview of the algorithm.

Algorithm 1 Pseudocode for Iterative Process

$x^0, x^{-1}=x^0, \beta^0=0, p_0=0, y^0=x^0, \alpha^k=\frac{1}{M}$
for $k=0,1,2,\ldots$ **do**
  Set $y^k=x^k+\beta^k(x^k-x^{k-1})$
  Set $x^{k+1}=y^k-x^k\Delta f(y^k)$
end
Define p_{k+1} to be the root in $[0,1]$ of the equation $1+p_{k+1}$
$(p_k^2-1)-p_{k+1}^2=0$
Set $\beta^{k+1}=p_{k+1}p_k^2$

20.52 Optimality of Nesterov's AGD

In the context of a convex function f that is M-smooth, it can be stated that the Accelerated Gradient Descent (AGD) algorithm stands out as the optimal choice among first-order algorithms for achieving

the fastest convergence rate. Moreover, no other algorithm that relies on computing the gradient differences $\Delta f(x_i)$ for every i from 1 to $k-1$ is capable of generating a sequence x_k that achieves a superior convergence rate compared to AGD. Consider the function $f(x) = \frac{1}{2}x^T Ax - {e_1}^T x$, where:

$$e_1^T = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}$$

and

$$A = \begin{bmatrix} 2 & -1 & 0 & \dots & \dots \\ -1 & 2 & -1 & \dots & \dots \\ 0 & -1 & 2 & -1 & \dots \\ \vdots & \vdots & \vdots & \vdots & -1 \end{bmatrix}$$

In the given context, $f(x) = \frac{1}{2}x^T Ax - {e_1}^T x$, where x is a vector, e_1 is a row vector with a leading 1 followed by zeros, and A is a symmetric matrix. The equation $Ax^* = e_1$ implies x^* as a solution with coordinates determined by $1 - \frac{i}{d+1} = x[i]$ for $i = 1, \dots, d$. Matrix A is symmetric with eigenvalues $\lambda_{\max}(A) \leqslant 4$, where the smallest eigenvalue is $\lambda_{\min} = 0$. This indicates that A is not strongly convex but is $M = 4$ smooth. Function f is both convex and $M = 4$ smooth, making it suitable for optimization. For any order algorithm, the update equation is $x^{k+1} = x^k + \sum_{j=0}^{k} \gamma^j \Delta f(x^j)$, using a step size γ to iteratively improve the solution.

Lemma 20.78. *For all iterations k, the vector x^k has non-zero entries only in its first k coordinates.*

Proof: We will prove this lemma by induction.For x^0, we have $\Delta f(x) = Ax - e$.For $k \geqslant 1$, when x^k is reached, it lies in the span of $\{e_1, e_2, \dots, e_k\}$, which means its non-zero entries are in the first k coordinates. By mathematical induction, we can deduce from the lemma that:

$$\|x^k - x^*\|^2 = \|x^*[k+1, \dots, d] - x^*[1, \dots k]\|^2 + \|x^*[k+1, \dots, d]\|^2$$

$$\|x^k - x^*\|^2 \geqslant \sum_{i=k+1}^{d} x^*[j]^2 = \sum_{i=k+1}^{d} (1 - \frac{j}{d+1})^2$$

A bit of algebra yields:

$$\|x^k - x^*\|^2 \geqslant \frac{1}{8}\|x^*\|^2 = \frac{1}{8}\|x^0 - x^*\|^2$$

Thus, we can conclude that:

$$f(x^k) - f(x^*) \geqslant \frac{3}{8(k+1)^2}\|x^0 - x^*\|^2$$

This inequality is a key result that demonstrates the convergence behavior of the algorithm. □

21.53 Conjugate Gradient

Let us look at Conjugate Gradient (C.G.) method for quadratic optimization problem:

$$\min_{x \in C} \quad f(x) = \frac{1}{2} x^T Q x - b^T x, \tag{126}$$

where $x \in \mathbb{R}^d$ and $\mathbb{Q}$ is a (symmetric) positive semi-definite matrix. This is an example of unconstrained optimization.

Algorithm 5: Conjugate Gradient

Data : Q, b

Initialize: Random vector x^0,

$p^0 = -(Qx^0 - b)$,

$r^0 = -p^0$

for k *in* $[0, 1, \cdots, d-1]$ **do**

  $x^{k+1} = x^k + \alpha^k p^k$

  $p^{k+1} = -r^{k+1} + \gamma^k p^k$

  where,

    $\alpha^k = \frac{{p^k}^T}{r^k}$

    $\gamma^k = \frac{{r^{k+1}}^T r^{k+1}}{{r^k}^T r^K}$

end

Output : Converged vector parameter x^T

Remark 21.79. *CG chooses* $\{p^k\}_{k=1}^{+\infty}$ *such that,* p^{k+1} *satisfies conjugate condition w.r.t* $p_0, p_1, \cdots, p_k$ *implying*

$$(p^j)^T Q p^{k+1} = 0 \; \forall j = 0, 1, \cdots k$$

Remark 21.80. *By construction* $\{p^k\}_{k=0}^{+\infty}$ *forms a linearly independent set.*

Remark 21.81. x^{k+1} *minimizes* f *over* $x^0 + span\{p_0, p_1, \cdots\}$ *implying CG terminates in at most* d *steps.*

21.54 Adaptive Gradient Methods

What we studied so far is constrained optimization:

$$f : \mathbb{R}^d \to \mathbb{R} \quad ; \quad \textit{convex}$$

$$\min_{x \in \Omega} f(x) \quad ; \quad \Omega : \text{closed and convex}$$

We can use:

1. Projected Gradient Descent;
2. Proximal Gradient Descent;
3. Frank Wolfe.

We want,

$$f(x^{N_\varepsilon}) - f(x^*) \leqslant \varepsilon \tag{127}$$

where, N_ε : Oracle complexity. One potential concern: **Choice of Step-Size**

Step-size	Structure of the Objective
$\alpha^k = \frac{1}{M}$	convex + M-smoothness
$\alpha^k = \frac{2}{M+m}$	m-strongly convex + M-smoothness
$\alpha^k \approx \frac{G}{\sqrt{k}}$	G- Lipschitz
$\alpha^k \approx \frac{c}{\sqrt{k}} or \frac{c}{k}$	c $\rightarrow$ *constant*

Remark 21.82. *(M, m, G, c) $\rightarrow$ are not available apriori and depends on the problem structure*

21.54.1 Motivation

Following are the motivations:

1. To automatically select the step size
2. Step size should adapt to the problem geometry (Ex. strongly convex, smoothness, etc.)

These properties are achieved by **Adaptive Gradient Descent**.

21.54.2 AdaGrad

Suppose that,

$$\max_{x,y\in\Omega} \|x-y\| \leqslant R \tag{128}$$

$$x \in \Omega \tag{129}$$

where R is the diameter of the constrained set Ω. For $k = 1, 2, \cdots, T$,

$$\begin{aligned} \alpha^k &= \frac{R}{\sqrt{\sum_{s=1}^k \|\nabla f(x^s)\|^2}} \\ x^{k+1} &= P_\Omega[x^k - \alpha^k \nabla f(x^k)], \end{aligned} \tag{130}$$

Returning the value,

$$\bar{x}^T = \frac{1}{T}\sum_{k=1}^T x^k \tag{131}$$

$$x^{k+1} = \arg\min_{u\in\Omega}\{f(x^k) + \langle \nabla f(x^k), u - x^k\rangle + \frac{1}{2\alpha^k}\|x^k - u\|^2\} \tag{132}$$

Exercise 21.83. *Show that*

$$x^{k+1} = \arg\min_{u\in\Omega}\{f(x^k) + \langle \nabla f(x^k), u - x^k\rangle + \frac{1}{2\alpha^k}\|x^k - u\|^2\}.$$

Convergence Analysis

Assumptions: We stipulate the assumption that $x^* \in \Omega$ is the unconstrained minima and consequently, $\nabla f(x^*) = 0$.

Theorem 21.84. *If f is convex and G-Lipschitz, i.e., $\|\nabla f(x)\| \leqslant G \quad \forall x \in \Omega$). Then running AdaGrad for T-steps achieves*

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{3RG}{\sqrt{T}},$$

where $\bar{x}^T = \frac{1}{T}\sum_{k-1}^{T} x^*$.

Remark 21.85. *A few remarks are in order:*

- *No step-size selection.*
- *Not dependant on initial conditions.*

22.55 Convergence Analysis for AdaGrad

Theorem 22.86. *Suppose f is convex and G-Lipschitz (which implies that $\|\nabla f\| \leqslant G$) we run AdaGrad for T steps, we obtain*

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{3RG}{\sqrt{T}}$$

Remark 22.1 Same convergence rate as gradient descent, Advantage is stepsize selection is independent of G.

Proof: Using convexity we get

$$f(\bar{x}^T) - f(x^*) \leqslant f(\tfrac{1}{T}\textstyle\sum_{i=1}^T x^k) - f(x^*).$$

Using Jensen inequality gives us

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{1}{T}\sum_{i=1}^T f(x^k) - \frac{1}{T}\sum_{i=1}^T f(x^*) \implies f(\bar{x}^T) - f(x^*) \leqslant \frac{1}{T}\sum_{i=1}^T [f(x^k) - f(x^*)]$$

We again use convexity,

$$f(x^k) - f(x^*) \leqslant \left\langle \nabla f(x^k), x^* - x^k \right\rangle \implies f(\bar{x}^T) - f(x^*) \leqslant \frac{1}{T}\sum_{i=1}^T \left\langle \nabla f(x^k), x^* - x^k \right\rangle$$

Now $x^{k+1} = P_\Omega[x^k - \alpha^k \nabla f(x^k)]$, we can use Minimum Principle

$$\langle P_\Omega(z) - z, P_\Omega(z) - u \rangle \leqslant 0 \text{ for all } u \in \Omega \tag{133}$$

and we have $z = x^k - \alpha^k \nabla f(x^k)$. To see this, $x^{k+1} = \arg\min_{u\in\Omega} \phi(u)$. So, x^{k+1} should satisfy the first order necessary condition (FONC):

$$\left\langle \nabla \phi(x^{k+1}), x^{k+1} - u \right\rangle \leqslant 0 \quad \forall u \in \Omega.$$

We choose $x^* = u$ (by Assumption $x^* \in \Omega$). We get

$$\left\langle \nabla f(x^k) + \frac{1}{\alpha^k}(x^{k+1} - x^k), x^{k+1} - x^* \right\rangle \leqslant 0 \implies \left\langle \nabla f(x^k), x^{k+1} - x^* \right\rangle \leqslant \frac{1}{\alpha^k}\left\langle x^k - x^{k+1}, x^{k+1} - x^* \right\rangle$$

We use $(a+b)^2 = a^2 + b^2 + 2ab$ which gives

$$\left\langle \nabla f(x^k), x^{k+1} - x^* \right\rangle \leqslant \frac{1}{2\alpha^k}[\|x^k - x^*\|^2 - \|x^k - x^{k+1}\|^2 - \|x^{k+1} - x^*\|^2] \tag{134}$$

We have

$$\left\langle \nabla f(x^k), x^k - x^* \right\rangle = \left\langle \nabla f(x^k), x^{k+1} - x^* \right\rangle + \left\langle \nabla f(x^k), x^k - x^{k+1} \right\rangle$$

we have

$$\leqslant \frac{1}{2\alpha^k}[\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2] - \frac{1}{2\alpha^k}\|x^k - x^{k+1}\|^2 + \left\langle \nabla f(x^k), x^k - x^{k+1} \right\rangle$$

Using Young's Inequality (Generalized Cauchy Schwarz), $ab \leqslant \frac{\lambda a^2}{2} + \frac{b^2}{2\lambda}; \lambda > 0$

$$\left\langle \nabla f(x^k), x^k - x^{k+1} \right\rangle \leqslant \tfrac{\lambda}{2}\|\nabla f(x^k)\|^2 + \tfrac{1}{2\lambda}\|x^k - x^{k+1}\|^2 - \tfrac{1}{2\alpha^k}\|x^k - x^{k+1}\|^2$$

We choose $\lambda = \alpha^k$

$$\left\langle \nabla f(x^k), x^k - x^{k+1} \right\rangle \leqslant \frac{\alpha^k}{2}\|\nabla f(x^k)\|^2 \tag{135}$$

We then write

$$\left\langle \nabla f(x^k), x^k - x^{k+1} \right\rangle \leqslant \frac{1}{2\alpha^k}[\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2] + \frac{\alpha^k}{2}\|\nabla f(x^k)\|^2$$

This implies

$$\begin{aligned}
\sum_{i=1}^{T} \left\langle \nabla f(x^k), x^k - x^{k+1} \right\rangle &\leqslant \sum_{i=1}^{T} \frac{1}{2\alpha^k}[\|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2] + \sum_{i=1}^{T} \frac{\alpha^k}{2}\|\nabla f(x^k)\|^2 \\
&\leqslant [\sum_{i=1}^{T}(\frac{1}{2\alpha^k} - \frac{1}{2\alpha^{k-1}}) + \frac{1}{2\alpha}]R^2 + \sum_{i=1}^{T} \frac{\alpha^k}{2}\|\nabla f(x^k)\|^2 \\
&\leqslant \frac{1}{2\alpha^T}R^2 + \sum_{i=1}^{T} \frac{\alpha^k}{2}\|\nabla f(x^k)\|^2
\end{aligned}$$

Recall, $\alpha^k = \frac{R}{\sqrt{\sum_{i=1}^{T}\|\nabla f(x^i)\|^2}}$ which gives us

$$\sum_{i=1}^{T} \frac{\alpha^k}{2}\|\nabla f(x^k)\|^2 = R\sum_{k=1}^{T} \frac{\|\nabla f(x^k)\|^2}{\sqrt{\sum_{i=1}^{T}\|\nabla f(x^i)\|^2}}.$$

Lemma 22.87. *If $a_1, a_2, \ldots\ldots a_T$ are positive numbers, then*

$$\sqrt{\sum_{i=1}^{T} a_i} \leqslant \sum_{t=1} \frac{a_t}{\sum_{s=1}^{T} a_s} \leqslant 2\sqrt{\sum_{i=1}^{T} a_i}.$$

Proof: Idea : $\int \frac{dx}{\sqrt{x}} = 2\sqrt{x}$ □

$$\sum_{k=1}^{T} \frac{\|\nabla f(x^k)\|^2}{\sqrt{\sum_{i=1}^{T}\|\nabla f(x^i)\|^2}} \leqslant 2\sqrt{\sum_{k=1}^{T}\|\nabla f(x^k)\|^2}$$

We have,

$$\sum_{i=1}^{T}\left\langle \nabla f(x^k), x^k - x^{k+1}\right\rangle \leqslant \sqrt{\sum_{k=1}^{T}\left\|\nabla f(x^k)\right\|^2} + 2\sqrt{\sum_{k=1}^{T}\left\|\nabla f(x^k)\right\|^2}$$

$$\sum_{i=1}^{T}\left\langle \nabla f(x^k), x^k - x^{k+1}\right\rangle \leqslant 3RG\sqrt{T}$$

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{1}{T}\sum_{i=1}^{T}\left\langle \nabla f(x^k), x^k - x^{k+1}\right\rangle \leqslant \frac{1}{T}3RG\sqrt{T}$$

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{3RG}{\sqrt{T}},$$

which completes the proof. □

23.56 AddaGrad

In this section we describe the adaptive gradient algorithm AdaGrad and provide a guarantee of the convergence rate obtained by it for convex and smooth functions. Recall that in the last lecture, we provided a guarantee of the convergence rate obtained by it for convex and G-Lipschitz functions.

Algorithm:

At time k,

$$x^{k+1} = P_{\Omega}[x^k - \alpha^k \nabla f(x^k)]$$

$$\alpha^k = \frac{R}{\sqrt{\sum_{s=1}^{k} \| \nabla f(x^s) \|^2}}$$

Run it for T rounds, and return $\bar{x}^T = \frac{1}{T}\sum_{k=1}^{T} x^k$.

Last Lecture:

For a function f that is convex and G Lipschitz, we showed that

$$f(\bar{x}^T) - f(x^*) \leqslant O\left(\frac{RG}{\sqrt{T}}\right)$$

Recall that in the above expression, x^* is unconstrained minima, i.e., $x^* \in \arg\min_{x \in \mathbb{R}^d}$. Also, $x^* \in \Omega$.

An Intermediate Result from the Last Lecture:

$$\sum_{k=1}^{T} \langle \nabla f(x^k), x^k - x^* \rangle \leqslant 3R \sqrt{\sum_{k=1}^{T} \| \nabla f(x^k) \|^2} \tag{136}$$

Recall that we obtain (136) using only convexity property of function f.

Convergence Guarantee:

For convex and M-smooth functions, AdaGrad attains $O\left(\frac{1}{T}\right)$ convergence rate, which is the same as obtained by GD, however, the step size selection does not require knowledge of smoothness parameter M.

Theorem 23.88. *Suppose f is convex and M-smooth and we run AddaGrad for T rounds, we obtain,*

$$f(\bar{x}^T) - f(x^*) \leqslant \frac{9}{2}\frac{MR^2}{T},$$

where, x^* is the unconstrained minima and $x^* \in \Omega$.

Proof: f is M-smooth implies

$$\| \nabla f(x) - \nabla f(y) \| \leqslant M \| x - y \|$$

f is convex and M-smooth implies (see HW 1)

$$f(y) \geqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2M} \| \nabla f(x) - \nabla f(y) \|^2 \tag{137}$$

The first two terms in RHS of the above expression is due to the convexity of f and the last term is extra gain from smoothness of f. Using (137) with $y = x^k, x = x^*$, we get,

$$f(x^*) \geqslant f(x^k) + \langle \nabla f(x^k), x^* - x^k \rangle + \frac{1}{2M} \| \nabla f(x^k) \|^2$$

In the above expression, x^* is unconstrained minima and hence implies $\nabla f(x^*) = 0$. Rearranging it we get,

$$f(x^k) - f(x^*) \leqslant \langle \nabla f(x^k), x^k - x^* \rangle - \frac{1}{2M} \| \nabla f(x^k) \|^2 \tag{138}$$

Now, we have all the results needed for the proof.

$$\begin{aligned}
f(\bar{x}^T) - f(x^*) &= f\left(\frac{1}{T}\sum_{k=1}^{T} x^k\right) - f(x^*) \\
&\leqslant^{(I1)} \frac{1}{T}\sum_{k=1}^{T}\left[f(x^k) - f(x^*)\right] \\
&\leqslant^{(I2)} \frac{1}{T}\sum_{k=1}^{T}\left[\langle \nabla f(x^k), x^k - x^* \rangle - \frac{1}{2M} \| \nabla f(x^k) \|^2\right] \\
&\leqslant^{(I3)} \frac{1}{T}\left[3R\sqrt{\sum_{k=1}^{T} \| \nabla f(x^k) \|^2} - \frac{1}{2M}\sum_{k=1}^{T} \| \nabla f(x^k) \|^2\right] \\
&\leqslant^{(I4)} \frac{1}{T}\left[9R^2M - \frac{9}{2}\frac{R^2M^2}{M}\right] \\
&= \frac{1}{T}\frac{9}{2}MR^2
\end{aligned}$$

The inequality $(I1)$ follows using Jensen inequality. The inequality $(I2)$ follows from (137) and inequality $(I3)$ follows from (136). The inequality $(I4)$ follows since the term obtained after $(I3)$ is a concave function in $\sqrt{\sum_{k=1}^{T} \| \nabla f(x^k) \|^2}$. □

Remark 23.89. *AdaGrad provides $O\left(\frac{1}{T}\right)$ convergence rate for convex and smooth functions without having knowledge of smoothness parameter M.*

Remark 23.90. *AdaGrad provides convergence rate guarantee on average iterate $\frac{1}{T}\sum_{k=1}^{T} x^k$, however, GD provides convergence rate guarantee on the final iterate x^T. Hence, GD gives a stronger guarantee.*

23.57 AdaGrad+ [2]

In the last section, we have seen the convergence of AdaGrad to an unconstrained minima $x^* \in \arg\min_{x\in\mathbb{R}^d} f(x)$ when $x^* \in \Omega$. In this section, we study the convergence of AdaGrad to a constrained minimum $x_c^* \in \arg\min_{x\in\Omega} f(x)$. Note that, in this case, $x^* \in \arg\min_{x\in\mathbb{R}^d} f(x)$ not lies inside Ω.

Remark 23.91. *It is worth noting that*

$$\begin{aligned}\nabla f(x^*) &= 0.\\ \nabla f(x_c^*) &\neq 0,\\ -\nabla f(x_c^*) &\in \mathcal{N}_\Omega(x_c^*).\end{aligned}$$

Remark 23.92. *The step size of AdaGrad will change. As we approach x_c^*, the gradient norm does not go to zero, however, $\| x^{k+1} - x^k \|$ goes to zero. We use this intuition to define step size α^k.*

Algorithm: AdaGrad+ [Alina Ene et.al 2020]

$$x^{k+1} \in \arg\min_{u\in\Omega}\left\{ f(x^k) + \langle \nabla f(x^k), u - x^k\rangle + \frac{1}{2\alpha^k}\| u - x^k \|\right\}$$

where, α^k satisfies,

$$\frac{1}{(\alpha^k)^2} = \frac{1}{(\alpha^{k-1})^2}\left[1 + \frac{\| x^k - x^{k-1} \|^2}{R^2}\right]$$

Remark 23.93. *We have,*

$$\alpha^k = \frac{R}{\sqrt{\frac{R^2}{(\alpha^1)^2} + \sum_{s=1}^{k-1}\frac{1}{(\alpha^s)^2}\| x^{s+1} - x^s \|^2}}$$

Remark 23.94. *If $\Omega = \mathbb{R}^d$, we can substitute*

$$\| x^{s+1} - x^s \|^2 = (\alpha^s)^2 \| \nabla f(x^s) \|^2$$

and get back the AdaGrad step size.

Remark 23.95. *If Ω is large, we can substitute*

$$\| x^{s+1} - x^s \|^2 \approx (\alpha^s)^2 \| \nabla f(x^s) \|^2$$

and get back the AdaGrad step size.

Convergence Guarantee:

Theorem 23.96. *f convex and G-Lipschitz, then we have the following guarantee:*

$$f(\bar{x}^T) - f(x_c^*) \leqslant O\Big(\frac{1}{\sqrt{T}}\Big)$$

Theorem 23.97. *f convex and M-smooth, then we have the following guarantee:*

$$f(\bar{x}^T) - f(x_c^*) \leqslant O\Big(\frac{1}{T}\Big)$$

For proof of Theorem 23.96 and Theorem 23.97, see Alina et.al, 2020.

AdaGrad Applications:

1. Classification (Image Net)
2. Language Model
3. RL-Game (Alpha Go)

AdaGrad Family:

- AdaGrad (Duchi et.al, 2010)
- AdaDelta[10] (Zeiler, 2012)
- RMS Prop (Momentum on top of gradient, 2014)
- Adam[4] (Kingma and Ba, 2015). The algorithm is based on adaptive momentum estimation. The paper is the most cited paper ever as of Nov. 2023 with a citation count of 158950.
- Adanam (Kingma and Ba)
- Nadam (Nesterov+ Adam)

Last class:

In the last class, we discussed Adagrad and its implementations for Convex + Smooth functions, Constrained Optimization, Convex + Lipschitz functions and verified guarantees for ADAM-Momentum based algorithm + Adaptivity .

24.58 Adaptive Movement Estimation algorithm (ADAM)

Before Proceeding to sequences let us define some new notation. For any matrix

$$Z = \begin{bmatrix} z_1 & z_2 & \dots & z_D \end{bmatrix}^\top, Y = \begin{bmatrix} y_1 & y_2 & \dots & y_D \end{bmatrix}^\top,$$

we define

$$Z \odot Y = [z_i y_j \delta_{ij}] = \begin{bmatrix} z_1 y_1 & z_2 y_2 & \dots & z_D y_D \end{bmatrix}^T$$

where $\odot$ denotes element-wise product. We also write

$$Z^{-\frac{1}{2}} = \left[\frac{1}{\sqrt{z_i}}\right]^T = \begin{bmatrix} \frac{1}{\sqrt{z_1}} & \frac{1}{\sqrt{z_2}} & \dots & \frac{1}{\sqrt{z_d}} \end{bmatrix}^T,$$

where $z_1, z_2 \dots z_d > 0$. Consider three sequences as follows,

Sequence Type	Sequence
Momentum sequence	$m^k = \beta^{(1)} m^{k-1} + (1-\beta^{(2)}) \nabla f(x^k)$, where $\beta^{(1)} \in (0,1)$
Intermediate sequence	$v^k = \beta^{(2)} v^{k-1} + (1-\beta^{(2)})[\nabla f(x^k) \odot \nabla f(x^k)]$
Iterate sequence	$x^{k+1} = x^k - \alpha^k (v^k)^{-\frac{1}{2}} \odot m^k$

Remark 24.1 In ADAM we only have to choose the initialization parameters namely $\beta^{(1)}$, $\beta^{(2)}$, α^k. The choice of these are not dependent upon the problem statement (objective function) or its properties. So hyper parameter tuning is quite faster as compared to SGD/GD.

24.58.1 Non-convergence of ADAM

It was shown in an ICLR-2018 research paper [8] that a counter example existed in which case ADAM will not converge the example objective function was,

$$\min_{x \in \mathbb{R}^d} F(x) \text{ , where } F(x) = \sum_{j=1}^{n} F_j(x) \text{ and } F_j(x) = \begin{cases} 5.5x^2 & \text{if } j = 1 \\ -0.5x^2 & \text{if } j \neq 1 \end{cases}$$

Therefore,

$$F(x) = (5.5 - (n-1)0.5)x^2 \text{ where n is a positive integer.}$$

if $n < 12$ the function $F(x)$ is strongly convex if $n = 12$ the function is convex and if $n > 12$ the function is concave hence, in general, the function is neither concave nor convex.
In [8] they have used various advanced mathematical tools to prove their arguments which we will not do here rather just take the result as is for generality sake .

Remark 24.2 Despite the non-convergent property of ADAM it is still used in industry for many problems since it only fails in specially crafted functions which are rare in real world scenarios.

24.59 AMSGrad - The Generalized ADAM

AMSGrad also uses updates like ADAM just with slight modifications and 'fixes'. Let us Define a new Projection operator type as

$$P_{\Omega,A}(y) = \arg\min_{x\in\Omega} \left\| A^{\frac{1}{2}}(y-x)^2 \right\| = \arg\min_{x\in\Omega} [(y-x)^T A(y-x)].$$

This is known as the *Mahalanobis norm*, where $A^{\frac{1}{2}}$ is the square-root of a positive-semi definite matrix A.

Sequence Type	Sequence
Momentum sequence	$m^k = \beta^{(1)} m^{k-1} + (1-\beta^{(2)})\nabla f(x^k)$, where $\beta^{(1)} \in (0,1)$
Intermediate sequence (1)	$v^k = \beta^{(2)} v^{k-1} + (1-\beta^{(2)})[\nabla f(x^k) \odot \nabla f(x^k)]$
Intermediate sequence (2)	$\hat{V}^k = \text{diag}(\hat{v}^k)$ where $\hat{v}^k = \max\{v^k, \hat{v}^{k-1}\}$
Iterate sequence	$x^{k+1} = P_{\Omega,A}[x^k - \alpha^k (v^k)^{-\frac{1}{2}} \odot m^k]$

This version of the ADAM converges for a larges class of loss functions, stochastic noise etc.

24.60 Optimization in Neural Networks (NNs)

Early versions and implementations of neural networks dealt with universal function approximators. Modern versions deal with the Classifications of data to predict patterns but the core idea behind the models remains the same.
Some examples of Linear Classifiers

1. Perceptrons
2. Support Vector Machines (SVMs)
3. Others ...

Some Examples of Non-Linear Classifiers

1. K-nearest Neighbours
2. Random Forest
3. Decesion Tress
4. Kernal-SVMs
5. Others ...

For Linear Classification **Fig 24.1** we have, $\{X_i, Y_i\}_{i=1}^N$ these are the collection of data points in a multi-dimensional space, where $X_i \in \mathbb{R}^d$ and $Y_i \in \{+1, -1\}$. Thus $\hat{Y} = \text{sign}(w^T X)$ where $\hat{Y}$ is the predicted value from the NN model and w is the hyperplane dividing the data points. Hence it is easy to observe

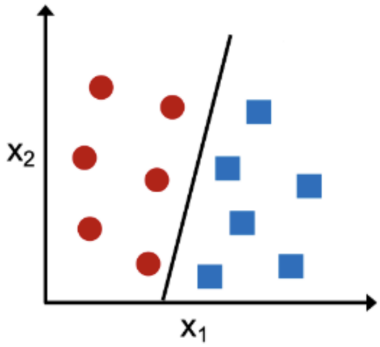


Figure 24.14. Linear Classification for d=2

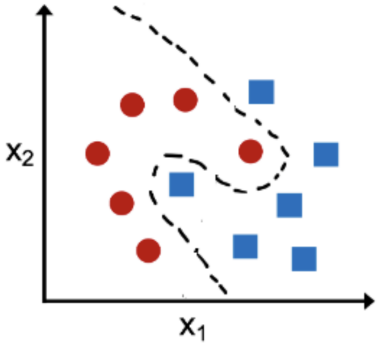


Figure 24.15. Non-Linear Classification for d=2

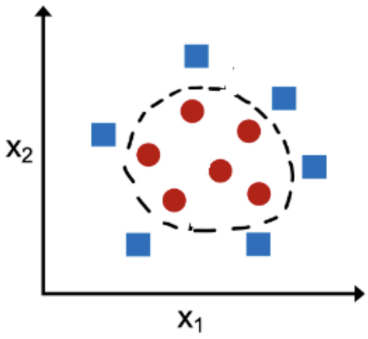


Figure 24.16. Non-Linear Classification for d=2

that predicting the geometries of the curve separating in **Fig 24.2/3** would be computationally much harder. Generalized/Validation guarantees are much better for NNs. We feed our data into the NNs and give out its prediction function for $\{X_i, Y_i\}_{i=1}^{N}$ as $\hat{y} = \hat{f}_W(x)$ where W is the predicted weight matrix. The error between predicted and actual value is $\varepsilon_{\text{error}} = \mathsf{E}_{(x,y)} l(y, \hat{f}_{W^*}(x))$ and we define

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, \hat{y})$$

where $l(\cdot)$ is the distance norm between two data points (usually euclidean). Now we minimize the objective function $W^* = \arg\min_{w \in \mathbb{W}} L(w)$, which will be commenced in detail in the next lecture.

Lecture 25 — November 3

Lecturer: Avishek Ghosh | *Scribe: Souvik Das*

25.61 Topics covered in the previous class

In the last few classes, we covered several adaptive algorithms, including AdaGrad, ADAM, and AMS-Grad. Toward the end of the class, a brief introduction to neural networks was provided.

Today's agenda:

We show that if for an m hidden node shallow neural network with ReLU activation and n training data, as long as m is large enough (in the over-parametrized regime) and no two inputs are parallel, and the parameters of the network are initialized randomly (according to a Gaussian random variate), the gradient descent converges to a globally optimal solution for the quadratic loss function. The primary reference used for this lecture is the article [1].

25.62 Introduction to neural networks

Neural networks (NNs) are a means of doing machine learning, in which a computer learns to perform some task by analyzing training examples. Usually, the examples have been hand-labeled in advance, for example, a recognition system based on NNs. A typical NN consists of an input layer, d hidden layers, and the output layer. Neural networks, as sophisticated black box models, excel in serving as powerful

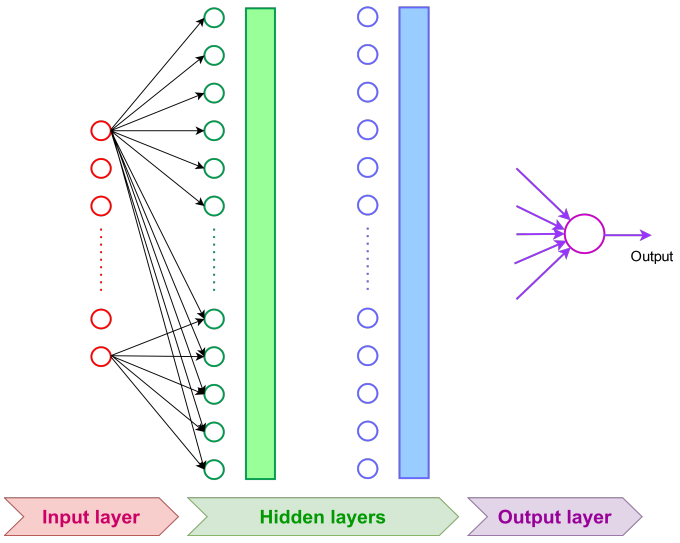


Figure 25.17. A two hidden layer neural network

tools for function approximation. Within their intricate architecture, they operate as nonlinear classifiers,

allowing them to capture complex relationships and patterns in data that traditional linear models might overlook.

Neural networks trained by first-order methods have achieved a remarkable impact on many applications, but their theoretical properties are still mysteries: randomly initialized first-order methods like stochastic gradient descent always find a global minimum even though the cost is non-convex or non-smooth. This lecture will demystify this for 2-layer neural network with ReLU activation layer.

In the sequel we focus on a 2-layer NN (consisting of 1-hidden layer). We will focus on ReLU activation function. A 2-layer NN is written as a function

$$\hat{y} \coloneqq f(\mathbf{w}, \alpha, x) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} \alpha_r \sigma(\mathbf{w}_r^\top x), \tag{139}$$

along with the following data:

- $x \in \mathbb{R}^d$ is the input to the NN;
- m denotes the number of hidden nodes;
- $\mathbf{w}_r \in \mathbb{R}^d$ is the weight vector of the r^{th} node. Combining all the nodes we adopt $\mathbf{w} \coloneqq (\mathbf{w}_1 \ \cdots \ \mathbf{w}_m)$;
- $\sigma(\cdot)$ corresponds to the ReLU activation function and is defined by $\sigma(y) \coloneqq \max(y, 0)$;
- $(\alpha_r)_{r=1}^m$ is a real-valued sequence of output weights.

Our desiderata is to employ empirical risk minimization (or training) with a quadratic loss. Given a training data set $(x_i, y_i)_{i=1}^n$, we want to minimize the quadratic loss function

$$L(\mathbf{w}, \alpha) = \frac{1}{2} \sum_{i=1}^{n} \big(f(\mathbf{w}, \alpha, x_i) - y_i\big)^2$$

In practice, one can consider the function f to be a logistic loss function or a cross-entropy loss function.

We fix the output layer (second layer) and apply gradient descent (GD) to optimize the first layer

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta \frac{\partial L(\mathbf{w}^k, \alpha)}{\partial \mathbf{w}},$$

where $\mathbf{w}^k = (\mathbf{w}_1^k \ \mathbf{w}_2^k \ \cdots \ \mathbf{w}_m^k)$, $\eta > 0$ is the step-size, and for each $r = 1, 2, \ldots, m$, we have

$$\mathbf{w}_r^{k+1} = \mathbf{w}_r^k - \eta \frac{\partial L(\mathbf{w}^k, \alpha)}{\partial \mathbf{w}_r}.$$

In other words, we have to compute

$$\begin{aligned} \frac{\partial L(\mathbf{w}^k, \alpha)}{\partial \mathbf{w}_r} &= \sum_{i=1}^{n} \big(f(\mathbf{w}^k, \alpha, x_i) - y_i\big) \frac{\partial f(\mathbf{w}^k, \alpha, x_i)}{\partial \mathbf{w}_r} \\ &= \frac{1}{\sqrt{m}} \sum_{i=1}^{n} \big(f(\mathbf{w}^k, \alpha, x_i) - y_i\big) \alpha_r x_i 1_d(\mathbf{w}_r^\top x \geqslant 0) \end{aligned}$$

Though this is only a shallow fully connected neural network, the objective function is still non-smooth and non-convex due to the use of the ReLU activation function. Even for this simple function, why a randomly initialized first-order method can achieve zero training error is not known.

Our goal:

We show that if m is large enough, i.e., $m \gg n$ (the number of hyper-parameters is much larger than the number of data points), and if we initialize the gradient descent through *Gaussian* random variables, then the gradient descent achieves zero training loss at an exponential speed, i.e., for every $\varepsilon > 0$, it finds a solution $\mathbf{w}^k$ such that $L(\mathbf{w}^k, \alpha) \leqslant \varepsilon$ in $k = O(\log \frac{1}{\varepsilon})$ iterations.

Assumption 25.98. *Define the matrix $H^\infty \in \mathbb{R}^{n\times n}$ with $H^\infty_{ij} = \mathsf{E}_{\mathbf{w}\sim\mathcal{N}(0,\mathbb{I})}[x_i^\top x_j 1_{\mathbf{w}^\top x_i \geqslant 0, \mathbf{w}^\top x_j \geqslant 0}]$. We assume that $\lambda_0 := \lambda_{\min}(H^\infty) > 0$.*

The following result summarizes the idea:

Theorem 25.99. *Suppose Assumption 25.98 holds. Set $m \geqslant C1\frac{n^6}{\lambda_0^4\delta^3}$ and the step size $\eta \leqslant \frac{C_2}{n^2},$. Let us initialize $\mathbf{w}_r \sim \mathcal{N}(0,\mathbb{I})$ and $\alpha_r \sim \mathsf{Unif}[-1,1]$ in i.i.d fashion. Then with probability greater than $1-\delta$, we have*

$$\left\|u^k - y\right\|^2 \leqslant \left(1 - \frac{\eta}{2}\right)^k \left\|x^0 - y\right\|^2,$$

where $u^k = (u_1^k\ u_2^k\ \cdots\ u_n^k) \in \mathbb{R}^{nd}$ with $u_i^k = f(\mathbf{w}^k, \alpha, x_i)$, and $y = (y_1\ y_2\ \cdots\ y_n) \in \mathbb{R}^n$. Here $C_1, C_2 >$ are universal constants.

Sketch of proof: We employ mathematical induction to prove the assertion of the theorem. Consider the base case. Put $k = 0$. Then $\left\|u^0 - y\right\|^2 = \left\|u^0 - y\right\|^2$, and the assertion hold for $k = 0$. Now we have

$$\begin{aligned}
\left\|u^{k+1} - y\right\|^2 &= \left\|u^{k+1} - u^k + u^{k+1} - y^2\right\| \\
&= \left\|u^k - y\right\|^2 + \left\|u^{k+1} - u^k\right\|^2 - 2(y - u^k)^\top (u^{k+1} - u^k) \\
&\leqslant \left(1 - \frac{\eta}{2}\right)^k \left\|u^0 - y\right\|^2 + \text{term 1} + \text{term 2}.
\end{aligned}$$

Let us focus our attention on Term 1. The following lemma is useful:

Lemma 25.100. $\left\|u^{k+1} - u^k\right\|^2 \leqslant \eta^2 n^2 \left\|u^k - y\right\|^2$.

Sketch of the proof of Lemma 25.92. Note that

$$u_i^{k+1} - u_i^k = \frac{1}{\sqrt{m}} \sum_{r=1}^m \alpha_r \{\sigma\big((\mathbf{w}_r^{k+1})^\top x_i\big) - \sigma\big((\mathbf{w}_r^k)^\top x_i\big)\},$$

and one obtain $\mathbf{w}_r^{k+1} = \mathbf{w}_r^k - \eta \frac{\partial L}{\partial \mathbf{w}_r^k}$ (from propagation of ReLU).

Now let us focus on Term 2. Again, the following lemma is useful:

Lemma 25.101. *If $\mathbf{w}_r^0 \sim \mathcal{N}(0, \mathbf{I_d})$. Then*

$$\left\|\mathbf{w}_r^{k+1} - \mathbf{w}_r^0\right\| \leqslant c_3 \sqrt{\frac{n}{m}} \left\|y - u^0\right\|$$

Observe that if $m \gg n$, the right-hand side of the above inequality is arbitrarily small. Using this it is possible to show that

$$(y-u^k)^\top (u^{k+1}-u^k) \geqslant (c+\text{small}) \left\|y-u\right\|^2.$$

Please refer to [1] for more details.

Combining everything and upon choosing an appropriate η, we get

$$\begin{aligned}
\left\|u^{k+1}-y\right\|^2 &\leqslant \left(1-\frac{\eta}{2}\right)^k \left\|u^0-y\right\|^2 + \eta^2 n^2 \left\|u^k-y\right\|^2 - 2(c+\text{small}) \left\|y-u\right\|^2 \\
&\leqslant \left(1-\frac{\eta}{2}\right)^k \left\|u^0-y\right\|^2 + \eta^2 n^2 \left\|u^k-y\right\|^2 \\
&\leqslant \left(1-\frac{\eta}{2}\right)^{k+1} \left\|u^0-y\right\|^2.
\end{aligned}$$

Remark 25.102. *Please note that the difference of prediction from k to $k+1$ is in fact small. Moreover, a crucial step in the proof is Lemma 25.101, which asserts that the network must be* over-parameterized *for the gradient descent to converge globally at an exponential rate.*

Last class:

In the last class, we discussed a few convergence results for neural networks. This lecture will focus on *Neural Tangent Kernel*.

26.63 Neural Tangent Kernel (NTK)

ANNs have achieved phenomenal results in numerous areas of science and engineering, in particular machine learning. It it well-known that with sufficiently many hidden neurons in the network, ANNs can approximate any function with an arbitrary accuracy. But it is largely a mystery that whether the ANN optimization oracle converges or not. Indeed the loss surface of neural networks optimization problems is highly non-convex: it has a high number of saddle points which may slow down the convergence.

ANNs are similar to kernel methods in the sense that ANNs have good generalization properties in spite of their usual over-parametrization i.e., large networks can fit random labels with good test accuracy. In the infinite-width limit, ANNs have a Gaussian distribution described by a kernel and in the same limit, the behavior of ANNs during training is described by a related kernel, which we call the neural tangent network (NTK). This lecture will only contain a glimpse of NTK without any theoretical details; see [3] for the convergence and generalization results.

26.63.1 The setup and preliminaries

We only focus on ANN architectures that are fully-connected with layers numbered from 0 (*input layer*) to L (*output layer*), each containing $n_0, \ldots, n_L$ hidden layers/neurons, and a nonlinear activation function $\sigma : \mathbb{R} \longrightarrow \mathbb{R}$.[1]

We focus on ANN realization function

$$F^L : \mathbb{R}^p \longrightarrow \mathscr{F};\ \mathbb{R}^p \ni \theta \mapsto f_\theta \in \mathscr{F},$$

where p denotes total number of parameters and $\theta \in \mathbb{R}^p$ and $\mathscr{F}$ is a infinite dimensional function space. The effective dimension of the ANN is

$$p := n_0 n_1 + n_1 n_2 + \cdots + n_{L-1} n_L = \sum_{l=0}^{L-1} n_l n_{l+1}. \tag{140}$$

The corresponding weight matrices are $W^{(l)} \in \mathbb{R}^{n_l \times n_{l+1}}$ and bias vectors $b^{(l)} \in \mathbb{R}^{n_{l+1}}$ for $l = 0, \ldots, L-1$. In our setup, the parameters are initialized as iid Gaussian $\mathscr{N}(0,1)$.

Defining the network function f_θ: We define the network functions recursively. Let $f_\theta(x) := \hat{\alpha}^{(L)}(x;\theta)$, where the functions $\hat{\alpha}^{(l)}(\cdot;\theta) : \mathbb{R}^{n_0} \longrightarrow \mathbb{R}^{n_l}$ are called *preactivation function* and the functions $\alpha^{(l)}$:

[1] Certain regularity conditions on $\sigma(\cdot)$ are needed to prove the convergence results.

$\mathbb{R}^{n_0} \longrightarrow \mathbb{R}^{n_l}$ are known as the *activation* function. Starting from the 0^{th}-layer to L^{th}-layer, these defined by:

$$\begin{aligned} \alpha^{(0)}(x;\theta) &= x, \\ \hat{\alpha}^{(l+1)}(x;\theta) &= \frac{1}{\sqrt{n_l}} W^{(l)} \alpha^{(l)}(x;\theta) + \beta b^{(l)}, \\ \alpha^{(l)}(x;\theta) &= \sigma\big(\hat{\alpha}^{(l)}(x;\theta)\big), \end{aligned} \tag{141}$$

where $\sigma(\cdot)$ is applied entrywise and $\beta > 0$ allows us to tune the influence of the bias. The factors $\frac{1}{\sqrt{n_l}}$ are important for obtaining a consistent asymptotic behaviour of the neural network and the width of the hidden layers $n_1, \ldots, n_{L-1}$ goes to infinity; see [3, Remark 1] for a detailed discussion. We need some definitions to proceed further: For a fixed input distribution p^{in} on the input space $\mathbb{R}^{n_0}$ the function function space $\mathscr{F}$ is defined as

$$\mathscr{F} := \{f(\cdot) \mid f : \mathbb{R}^{n_0} \longrightarrow \mathbb{R}^{n_L}\},$$

and on $\mathscr{F}$ we define the inner product

$$\langle f, g \rangle_{p^{\text{in}}} := \mathbb{E}_{x \sim p^{\text{in}}}\big[\langle f(x), g(x)\rangle\big] = \mathbb{E}_{x \sim p^{\text{in}}}\big[f(x)^\top g(x)\big] \quad \text{for all } f, g \in \mathscr{F}. \tag{142}$$

We also assume that p^{in} is the emphirical distribution of a finite number of data points (x_i) for $i = 1, \ldots, N$ i.e., p^{in} can be thought of a sum of individual atoms at points or dirac measures $p^{\text{in}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_i}$.

26.63.2 Kernel gradient

Last time in class, to train a NN we constructed a cost function which was nonconvex. One of the main thrust of NTK is to make the cost function convex and subsequently the training will be tractable. In [3] it is shown that during training, the network function f_θ follows a descent along the kernel gradient with respect to the Neural Tangent Kernel (NTK) which makes it possible to study the training of ANNs in the function space $\mathscr{F}$, on which the cost L is convex. We need the following definitions:

Definition 26.103. *A* multi-dimensional kernel $K : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \longrightarrow \mathbb{R}^{n_L \times n_L}$ *is a function which maps* $(x_1, x_2) \in \mathbb{R}^{n_0} \times \mathbb{R}^{n_0}$ *to a matrix* $K(x_1, x_2) \in \mathbb{R}^{n_L \times n_L}$ *such that* $K(x_1, x_2) = K(x_2, x_1)^\top$.

Such a kernel defines the map $\langle \cdot, \cdot \rangle : \mathscr{F} \times \mathscr{F} \longrightarrow \mathbb{R}$ by

$$\langle f, g \rangle_K := \mathbb{E}_{x_1, x_2 \sim p^{\text{in}}}\big[f(x_1)^\top K(x_1, x_2) g(x_2)\big]. \tag{143}$$

Another important observation is to note that for a fixed $x_1 \in \mathbb{R}^{n_0}$ partial application of the kernel $K_{i,\cdot}(x_1, \cdot)$ is a function in $\mathscr{F}$.

Definition 26.104. *Given a vector space V over the field of real numbers $\mathbb{R}$, the dual space $V^\star$ corresponding to the primal space V is defined as the set of all linear functionals $f : V \longrightarrow \mathbb{R}$, i.e.,*

$$V^\star := \{f(\cdot) \mid f : V \longrightarrow \mathbb{R}, \text{ and } f(\cdot) \text{ is linear}\}$$

In our context the $\mathscr{F}$ is the primal vector space and we write the dual of $\mathscr{F}$ by $\mathscr{F}^\star$ which consists of linear functionals $\mu : \mathscr{F} \longrightarrow \mathbb{R}$ such that for some $h \in \mathscr{F}$ we define $\mu(f) := \langle h, f \rangle_{p^{\text{in}}}$ for all $f \in \mathscr{F}$. It is easy to check $\mu(\cdot)$ is indeed an bound linear function from $\mathscr{F}$ to $\mathbb{R}$ and thus a functional belonging to

$\mathscr{F}^\star$. We also define the *reverse link* function $\Phi_K : \mathscr{F}^\star \longrightarrow \mathscr{F}$ which maps a dual element $\mu(\cdot) = \langle h, \cdot \rangle$ to the function f_μ such that

$$f_\mu(x) = \Phi_K(\mu)(x) \coloneqq \langle h, K(x,\cdot)\rangle_{p^{\text{in}}}. \tag{144}$$

In our setting we have a finite data-set $x_1, \ldots, x_n \in \mathbb{R}^{n_0}$. The (functional) derivative of the cost L at a point $f_0 \in \mathscr{F}$ can be viewed as an element of $\mathscr{F}^\star$, which we write $\partial_f L|_{f_0}$. For some $h|_{f_0} \in \mathscr{F}$ we write

$$\partial_f L|_{f_0} = \left\langle h|_{f_0}, \cdot \right\rangle_{p^{\text{in}}}.$$

The ***kernel gradient*** is defined as

$$\nabla_K L|_{f_0} = \Phi_K(\partial_f L|_{f_0}) = \mathbb{E}_{x\sim p^{\text{in}}}\left[(f_0(x) - f^*(x))^\top K(\cdot, x)\right].$$

In terms of the data-set the kernel gradient can be written as

$$\nabla_K L|_{f_0}(x) \coloneqq \frac{1}{N}\sum_{j=1}^{N} K(x, x_j) h|_{f_0}(x_j).$$

A time dependent function $t \mapsto f(t)$ follows kernel GD update if it satisfies the differential equation

$$\partial_t f(t) = -\nabla_K L|_{f(t)}.$$

During kernel gradient descent the cost evolves as

$$\partial_t L_{f(t)} = -\left\langle h|_{f(t)}, \nabla_K L|_{f(t)} \right\rangle_{p^{\text{in}}} = \|h|_{f(t)}\|_K^2.$$

Convergence to a critical point of L is hence guaranteed if the kernel K is positive definite with respect to the norm $\|\cdot\|_{p^{\text{in}}}$: the cost is then strictly decreasing except at points such that $\left\|h|_{f(t)}\right\| = 0$. If the cost is convex and bounded from below, the function $f(t)$ therefore converges to a global minimum as $t \to +\infty$.

26.63.3 Neural tangent kernel

During training phase of the NN the network function f_θ evolves along the negetive kernel gradinet

$$\partial_t f_{\theta(t)} = \nabla_{\Theta^{(L)}} L|_{f_{\theta(t)}} \tag{145}$$

with respect to the ***Neural Tangent Kernel***

$$\Theta^{(L)}(\theta)(x_1, x_2) \coloneqq \sum_{p=1}^{P} \left(\partial_{\theta_p} F^{(L)}(\theta)(x_1)\right)^\top \left(\partial_{\theta_p} F^{(L)}(\theta)(x_2)\right)$$

which is

$$\Theta^{(L)}(\theta) \coloneqq \sum_{p=1}^{P} \partial_{\theta_p} F^{(L)}(\theta) \otimes \partial_{\theta_p} F^{(L)}(\theta)(x_2)$$

Bibliography

[1] S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[2] Alina Ene, Huy L Nguyen, and Adrian Vladu. Adaptive gradient methods for constrained convex optimization and variational inequalities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7314–7321, 2021.

[3] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] A. S. Nemirovskij and D. B. Yudin. Problem complexity and method efficiency in optimization. *SIAM Review*, 1983.

[6] Y. Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[7] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends®in Optimization*, 1(3):127–239, 2014.

[8] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[9] S. J. Wright and B. Recht. *Optimization for Data Analysis*. Cambridge University Press, 2022.

[10] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.