Optimization

Hasan Poonawala

Contents

Chapter 1

Introduction

1.1 Problem Formulation

A general optimization problem over n-dimensional vector x is of the form

$$\min_{x} \quad f_0(x) \tag{1.1}$$

$$s.t. \quad f_i(x) \leq b_i, \quad i \in \{1, \ldots, m\} \tag{1.2}$$

1.2 Least Squares and Linear Programming

1.2.1 Least Squares

$$\min_{x} \quad f_0(x) = \|Ax - b\|_2^2 = \sum_{i=1}^{k} (a_i^T x - b)^2, \tag{1.3}$$

where $A \in \mathbb{R}^{k \times n}$, and $b \in \mathbb{R}^k$.

The solution is

$$x^\star = (A^T A)^{-1} A^T b.$$

The complexity is $O(n^2 k)$, with a known constant.

1.2.2 Linear Program

$$\min_{x} \quad c^T x \tag{1.4}$$

$$s.t. \quad a_i^T x \leq b_i, \quad i \in \{1, \ldots, m\} \tag{1.5}$$

The complexity in practice is order $n^2 m$ (assuming $m \geq n$) but with a constant that is less well characterized than for least-squares.

1.3 Overview of Algorithms

- The optimization problem as stated (objective+constraints) is the *primal* problem.
- We may derive a *dual* problem which has a set of dual variables (one for each constraint). The solution of the dual problem bounds that of the primal.
- At the **heart of the matter** is the fact that the sub-gradients (covectors) at a point possibly tells you something about how taking a step in various directions (vectors) will change the objective (increase, decrease, or constant).

- Most primal approaches use the sub-gradients of the function to define a descent direction, and for a small enough step size the function at the end of the step is smaller.
- The KKT Conditions (differentiable case) identify the optimal point by stipulating that at such an optimal, you cannot find a descent direction for x, at least not one that will preserve constraints (necessary always; sufficient sometimes, often for convex problems).
- The same idea behind the KKT conditions lead to most non-primal approaches.
  - In some cases, the dual problem is easier to solve than the primal problem. Solving the dual problem leads to a solution of the primal problem, because of the nature of duality.
  - For constrained problems, descending is often the wrong choice from the current estimate. Distance to the feasible set is an important factor in updates, and proximal methods incorporate this observation.

Part I

Theory

Chapter 2

Convex Functions

2.1 Basic Proporties

2.1.1 Definition

A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is *convex* if $\mathbf{dom} f$ is a convex set and if for all $x, y \in \mathbf{dom} f$, and θ with $0 \leq \theta \leq 1$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y). \tag{2.1}$$

2.1.2 Extended-value extensions

Define

$$\tilde{f}(x) = \begin{cases} f(x), & \text{if } x \in \mathbf{dom} f \\ \infty, & \text{otherwise.} \end{cases} \tag{2.2}$$

2.1.3 First-order conditions

Suppose f is differentiable. Then f is convex if and only if $\mathbf{dom} f$ is convex and

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \tag{2.3}$$

holds for all $x, y \in \mathbf{dom} f$.

2.1.4 Second-order conditions

We now assume that f is twice differentiable. Then f is convex if and only if $\mathbf{dom} f$ is convex and and its Hessian is positive semidefinite: for all $x \in \mathbf{dom} f$,

$$\nabla^2 f(x) \succeq 0. \tag{2.4}$$

2.1.5 Sublevel Sets

2.1.6 Epigraph

The epigraph of f is given by

$$\mathbf{epi}\ f = \{(x, y) \colon x \in \mathbf{dom} f, f(x) \leq t\}. \tag{2.5}$$

The link between convex sets and convex functions is via the epigraph. A function is convex if and only if its epigraph is a convex set.

Redundant: We can convert a function $f\colon \mathbb{R}^n \to R$ into a set in $\mathbb{R}^n$ using the epigraph **epi-**f.

$$\text{epi} - f = \{(x, y)\colon y \geq f(x)\} \tag{2.6}$$

2.1.7 Indicator Function

We can convert a set S into a function using the indicator function $\mathbb{I}_S(x)$ given by

$$\mathbb{I}_S(x) = \begin{cases} 0, & \text{if } x \in S \\ \infty, & \text{otherwise.} \end{cases} \tag{2.7}$$

2.2 Operations that preserve convexity

- Nonnegative weighted sums
- Composition with an affine mapping
- Pointwise maximum and supremum
- Composition of certain types of functions
- minimization (certain kind)
- perspective of a function $g(x, t) = tf(x/t$.
- conjugation

2.3 The Conjugate Function

2.3.1 Definition

Let $f\colon \mathbb{R}^n \to \mathbb{R}$. The function $f^\star\colon \mathbb{R}^n \to \mathbb{R}$, defined as

$$f^\star(y) = \sup_{x \in \mathbf{dom} f} \left(y^T x - f(x)\right) \tag{2.8}$$

is called the *conjugate* function of f.

What it is. Consider the epigraph of f, which lies in $\mathbb{R}^n = \mathbb{R}^n \times \mathbb{R}$. Every dual vector y parametrizes a hyperplane through the origin in the same space as the epigraph, given by $(x, y^T x)$. Then, $f^\star(y)$ is the largest vertical (signed) distance between this hyperplane and $f(x)$. The solution to the equation $f^\star(y) = 0$ will therefore pick out the supporting hyperplanes to f that pass through the origin. Note that there are many supporting hyperplanes to f that do not pass through the origin.

Examples:

- Affine function $f(x) = ax + b$, $f^\star(a) = -b$, singleton domain
- Negative logarithm $f(x) = -\log x$, $f^\star(y) = -\log(-y) - 1$ for $y < 0$
- Exponential $f(x) = e^x$, $f^\star(y) = y\log(y) - y$ for $y > 0$
- Negative entropy $f(x) = x\log x$, $f^\star(y) = e^{y-1}$ for $y \in \mathbb{R}$
- Inverse $f(x) = 1/x$, $f^\star(y) = -2\sqrt{-y}$ for $y \leq 0$.

2.3.2 Basic Properties

Fenchel's Inequality.

$$f(x) + f^\star(y) \geq x^T y \tag{2.9}$$

$$\text{Proof: } f(x) + f^\star(y) = f(x) + \sup_{x' \in \mathbf{dom}\ f} y^T x' - f(x') \geq f(x) + y^T x - f(x) = y^T x \tag{2.10}$$

Conjugate of the conjugate. If f is convex and closed, then $(f^\star)^\star = f$

Proposition 1 (Nicholar Harvey)**.** *Suppose that f is closed and convex. Then the following are equivalent:*

$$y \in \partial f(x) \tag{2.11}$$
$$x \in \partial f^\star(y) \tag{2.12}$$
$$\langle x, y\rangle = f(x) + f^\star(y) \tag{2.13}$$

Proof. If $y \in \partial x$, then

$$\begin{aligned} & f(z) \geq f(x) + \langle z - x, y\rangle && \forall z \in \mathbf{dom}\ f && (2.14)\\ \implies & y^T x - f(x) \geq y^T z - f(z) && \forall z \in \mathbf{dom}\ f && (2.15)\\ \implies & y^T x - f(x) = f^*(y) && && (2.16)\\ \implies & f(x) + f^*(y) = y^T x && && (2.17) \end{aligned}$$

Given this result,

$$\begin{aligned} & f^*(z) \geq z^T x - f(x) && \forall z \in \mathbf{dom}\ f^* && (2.18)\\ & \quad = x^T z - x^T y + f^*(y) && (x^T y = f(x) + f^\star(y)) && (2.19)\\ & \quad = x^T(z - y) + f^*(y) && && (2.20)\\ \implies & f^*(z) \geq x^T(z - y) + f^*(y) && \forall z \in \mathbf{dom}\ f^* && (2.21)\\ \implies & x \in \partial f^\star(y) && && (2.22) \end{aligned}$$

Again,

$$\begin{aligned} \text{Conj. fn.} \quad & f^*(y) \geq y^T z - f(z) && \forall z \in \mathbf{dom}\ f && (2.23)\\ \implies & x^T y - f(x) \geq y^T z - f(z) && (x^T y = f(x) + f^\star(y)) && (2.24)\\ \implies & f(z) \geq y^T(z - x) + f(x) && \forall z \in \mathbf{dom}\ f && (2.25)\\ \implies & y \in \partial f(x) && && (2.26) \end{aligned}$$

□

Differentiable functions. The conjugate of a differentiable function f is called the *Legendre transform* of f. Suppose f is convex and differentiable, with $\mathbf{dom} f = \mathbb{R}^n$. Any maximizer $x^\star$ of $y^T x - f(x)$ satisfies $y = \nabla f(x^\star)$. Therefore,

$$f^\star(y) = (x^\star)^T \nabla f(x^\star) - f(x^\star). \tag{2.27}$$

So, if we can solve the equation $y = \nabla f(z)$ for z, then we can compute $f^\star(y)$.

Scaling and Composition with affine functions If $g(x) = f(Ax + b)$, with A non-singular, then

$$g^\star(y) = f^\star(A^{-T} y) - b^T A^{-T} y \tag{2.28}$$

with $\mathbf{dom} g^\star = A^T \mathbf{dom} f^\star$

Therefore, the Legendre transform is

$$f^\star(y) = z^T \nabla f(z) - f(z); \qquad \text{where } y = \nabla f(z) \tag{2.29}$$

Sums of independent functions Independence means functions of disjoint sets of variables.

$$f^\star(w, z) = f_1^\star(w) + f_2^\star(z) \tag{2.30}$$

2.4 Bregman Divergence

For a convex function f, its Bregman divergence is

$$D_f(x, y) = f(x) - f(y) - \langle x - y, \nabla f(y) \rangle \tag{2.31}$$

This divergence is the error in the first xorder approximation of f at x, defined by y.

In general, $D_f(x, y)$ is convex in x, but often not convex in y.

x

Lemma 2. *Let f be closed, convex and differentiable. Fix any $x, y \in$ **dom** f. Define $\hat{x} = \nabla f(x)$ and $\hat{y} = \nabla f(y)$. Then*

$$x = \nabla f^*(\hat{x}) \tag{2.32}$$
$$D_f(x, y) = D_{f^*}(\hat{y}, \hat{x}) \tag{2.33}$$

Proof. Note that by definition, $\hat{x} \in \partial f(x)$, $\hat{y} \in \partial f^(y)$ Since $\hat{x} = z = \nabla f(x)$, meaning $z \in \partial f(x)$,

$$f(x) + f^*(z) = z^T x \tag{2.34}$$
$$\implies \nabla f(x) + \nabla f^*(y) = \nabla_y (y^T x) \tag{2.35}$$
$$\implies \nabla f^*(\hat{x}) = x \tag{2.36}$$

For the second claim:

$$\begin{aligned} D_{f^*}(\hat{y}, \hat{x}) &= f^*(\hat{y}) - f^*(\hat{x}) - \langle \hat{y} - \hat{x}, \nabla f^*(\hat{x}) \rangle && (2.37) \\ &= \langle \hat{y}, y \rangle - f(y) - \langle \hat{x}, x \rangle + f(x) - \langle \hat{y} - \hat{x}, x \rangle && (2.38) \\ &= f(x) - f(y) - \langle \hat{y}, x - y \rangle && (2.39) \\ &= f(x) - f(y) - \langle \nabla f(y), x - y \rangle && (2.40) \\ &= D_f(x, y) && (2.41) \end{aligned}$$

□

The generalized Pythagoras Theorem is

$$D_f(x, y) + D_f(z, x) - D_f(z, y) \leq (\nabla f(x) - \nabla f(y))^T (x - z) \tag{2.42}$$

The Bregman projection is

$$\Pi_C^f(y) = \arg\min_{x \in C} D_f(x, y) \tag{2.43}$$
$$\implies 0 \in \nabla f(y) - \nabla f(x) + N_C^P(x) \tag{2.44}$$

Properties:

- Linearity: $D_{f_1 + \lambda f_2}(x, y) = D_{f_1}(x, y) + \lambda D_{f_2}(x, y)$
- Unaffected by linearity: $f_2(x) = f_1(x) + a^T x + b \implies D_{f_1} = D_{f_2}$.
- Gradient: $\nabla_x D_f(x, y) = \nabla f(x) - \nabla f(y)$

Chapter 3

Nonsmooth Functions

3.1 Sub-gradients

3.1.1 Definition

A sub-gradient of f at x is any vector g that satisfies

$$f(y) \geq f(x) + g^T(y - x) \quad \forall y \in \mathcal{D}$$

For f differentiable, g is the singleton set containing ∇f. For other types of functions, the subdifferential $\partial f(x)$ is

$$\partial f(x) = \{g \in \mathbb{R}^n \colon f(y) \geq f(x) + g^T(y - x) \forall y \in \mathcal{D}\} \tag{3.1}$$

Intuitively, g is a sub-gradient of f at x if it is the **projection** of a hyperplane that lies below the epi-graph **epi-**f of f.

- For differentiable functions, only the tangent plane does so. Any other plane 'enters' **epi-**f.
- For a function that is the max of hyper-planes, the subgradients at a line where two planes meet is

For non-differentiable functions, the sub-gradient is set-valued at points of non-differentiability.

3.1.2 Relationship To Conjugate Function

The conjugate function is $f^\star(y) = \sup_{x \in \mathbf{dom}\ f} y^T x - f(x)$. The sub-differential y at x satisfies

$$\begin{align} & f(z) \geq f(x) + y^T(z - x) \quad \forall z \in \mathbf{dom}\ f \tag{3.2} \\ \implies & f(z) - g^T z \geq f(x) - y^T x \tag{3.3} \\ \implies & \partial f(x) = \{y \colon f^\star(y) = y^T x - f(x)\} \tag{3.4} \end{align}$$

This derivation is by Thibaut Lienart

3.2 Proximal Operator

The *proximal operator*, also called *proximal point mapping*, $\mathbf{prox}_f \colon \mathbb{R}^n \to \mathbb{R}^n$ of f is defined by[1]

$$\mathbf{prox}_f(v) = \arg\min_x \left(f(x) + \frac{1}{2}\|x - v\|_2^2 \right) \tag{3.5}$$

[1] Notes by Benjamin Recht

The function minimized on the righthand side is strongly convex and not everywhere infinite, so it has a unique minimizer for every $v \in \mathbb{R}^n$.

Examples:

- Quadratic function $f(x) = \frac{1}{2}x^T Px + q^T x + r (P \succeq 0)$. $\mathbf{prox}_f(v) = (I + P)^{-1}(v - q)$.
- l_1 norm $f(x) = \|x\|_1$.

$$\mathbf{prox}_f(x) = \begin{cases} x_i - 1 & \text{if } x_i \geq 1 \\ 0 & \text{if } |x_i| \leq 1 \\ x_i + 1 & \text{if } x_i \leq -1 \end{cases} \quad \text{(Soft thresholding / Dead zoning)}$$

Note that $\mathbf{prox}_{\lambda f}(v) = \arg\min_x \left(f(x) + \frac{1}{2\lambda}\|x - v\|_2^2\right)$.
By the first order optimality conditions, we conclude that $\mathbf{prox}_f(x)$ is the unique point satisfying

$$x - \mathbf{prox}_f(x) \in \partial f(\mathbf{prox}_f(x)) \tag{3.6}$$

Equivalently, the proximal operator evaluates the subdifferential (From this source), though perhaps only for convex functions:

$$z = \mathbf{prox}_{\gamma f}(x) \iff \gamma^{-1}(x - z) \in \partial f(z).$$

These expressions arre a generalization of what we know for the projection operator to a set C (see Section 3.2.1): $x - \text{proj}_C(x) \in N_S^P(\text{proj}_C(x))$. The generalization says that if you take a step from x to $\mathbf{prox}_f(x)$, the reverse of the direction you moved in is a sub-gradient of f, at the new point. This idea may often allow you to derive a closed-form expression for the proximal operator, especially when δf is a singleton. Alternatively, a least-squares problem may provide a solution.

Lemma 3. *Let f be convex on X. Let $x, y \in X$, $g_x \in \partial f(x)$, $g_y \in \partial f(y)$, then*

$$\langle g_x - g_y, x - y\rangle \geq 0$$

Proof.

$$f(x) - f(y) \geq \langle g_x, x - y\rangle \tag{3.7}$$
$$f(y) - f(x) \geq \langle g_y, y - x\rangle \tag{3.8}$$
$$\implies 0 \geq \langle g_x - g_y, y - x\rangle \tag{3.9}$$
$$\implies 0 \leq \langle g_x - g_y, x - y\rangle \tag{3.10}$$

□

Lemma 4. *Let $Q_\nu(x) = x - \mathbf{prox}_{\nu f}(x)$, then*

i) $\nu^{-1}Q_\nu(x) \in \partial f(\mathbf{prox}_f(x))$

ii) $\langle \mathbf{prox}_{\nu f}(x) - \mathbf{prox}_{\nu f}(z), Q_\nu(x) - Q_\nu(z)\rangle \geq 0$

iii) $\|\mathbf{prox}_{\nu f}(x) - \mathbf{prox}_{\nu f}(z)\|^2 + \|Q_\nu(x) - Q_\nu(z)\|^2 \leq \|x - z\|^2$

iv) $\|x - z\| = \|\mathbf{prox}_{\nu f}(x) - \mathbf{prox}_{\nu f}(z)\|$ *if and only if* $x - z = \mathbf{prox}_{\nu f}(x) - \mathbf{prox}_{\nu f}(z)$

Proof. Claim i) follows from FOC. Claim ii) follows from i) combined with Lemma 3. Claim iii) follows from $x = \mathbf{prox}_{\nu f}(x) + Q_\nu(x)$ and expanding $\|x - z\|^2$ then using ii). Claim iv) follows from iii). □

Corollary 5 (Proximal operator is a contraction)**.**

$$\|\mathbf{prox}_{\nu f}(x) - \mathbf{prox}_{\nu f}(z)\|^2 \leq \|x - z\|^2$$

3.2.1 Projection to a Set

For any set S, and any $x \notin S$ we may define the projection operator

$$\text{proj}_S(x) = \arg\min_{s\in S} d(x,s) \tag{3.11}$$

Then, $x - \text{proj}_S(x)$ is the **projected normal vector** that points from $s = \text{proj}_S(x)$ to x. Let $s \in \partial S$. Then

$$N_S^P(s) = \{x - \text{proj}_S(x) \colon x \notin s, a = \text{proj}_S(x)\} \tag{3.12}$$

The set $N_S^P(a)$ is the *projected normal cone* to S at a.
$\mathbf{prox}_f(v) = \text{proj}_S(v)$ when $f = \mathbb{I}_S$.

3.2.2 Proximal Sub-differential

The proximal sub-differential $\partial_P f(x)$ of a function at x is a subset of dom$f$. It is the projection of the projected normal cone to the epigraph of f at $(x, f(x))$.

$$\partial_P f(x) = \{\zeta \in \mathbb{R}^n \colon (\zeta, -1) \in N^P_{\text{epi} f}(x, f(x))\} \tag{3.13}$$

As Clarke mentions,

Theorem 6 (F. Clarke [2],pg. 33)**.** *Let f be lower semi-continuous and $x \in$ **dom** f. Then $\zeta \in \partial_P f(x)$ if and only if $\exists\sigma > 0$, $\eta > 0$ such that*

$$f(y) \geq f(x) + \langle \zeta, y - x\rangle - \sigma\|y - x\|^2, \quad \forall y \in B(x;\eta) \tag{3.14}$$

A corollary: For a convex function, $\partial f(x) = \partial_P f(x)$

3.2.3 Resolvent

From Thibault Lienart's blog. Constrained minimization of $f(x)$ is simply unconstrained minmization of $g(x) = f(x) + \mathbb{I}_C$. The FOC then dictate that $x^\dagger$ is a minimizer of the constrained problem if $0 \in \partial f(x^\dagger) + N_C^P(x^\dagger)$. When $f(x) = \frac{1}{1}\|x - v\|^2$, then $x^\dagger$ is exactly $\mathbf{prox}_{\mathbb{I}_C}(v) = \text{proj}_C(v)$.

Going back to the FOC, when $f(x) = \frac{1}{1}\|x - v\|^2$, then $\partial f(x) = (x - v)$, so that the FOC becomes

$$\begin{aligned} 0 &\in x^\dagger - v + N_C^P(x^\dagger) && (3.15)\\ \implies v &\in x^\dagger + N_C^P(x^\dagger) && (3.16)\\ \implies v &\in \left(\text{id} + N_C^P\right)(x^\dagger) && (3.17)\\ \implies x^\dagger &= \left(\text{id} + N_C^P\right)^{-1}(v) && (3.18)\\ \implies \text{proj}_C &= \left(\text{id} + N_C^P\right)^{-1} && (3.19) \end{aligned}$$

This relationship is a special case of the resolvent of the subdifferential operator (due to FOC, or (3.6)):

$$\mathbf{prox}_{\lambda f} = (I + \lambda\partial f)^{-1} \tag{3.20}$$

3.2.4 Properties

1. If f is closed and convex, then $\mathbf{prox}_f(x)$ exists and is unique for all x.
2. Separable sum: If $f(x) = \sum f_i(x_i)$ then $\left(\mathbf{prox}_f(x)\right)_i = \mathbf{prox}_{f_i}(x_i)$
3. Fixed point: x^* minimizes f if and only if $x^* = \mathbf{prox}_f(x^*)$.
4. Conjugate (Moreau decomposition): $\mathbf{prox}_{tf^*}(x) = x - t\mathbf{prox}_{f/t}(x/t)$

Chapter 4

Duality

4.1 Lagrange Dual Function

4.1.1 The Lagrangian

Consider variable $x \in \mathbb{R}^n$

$$\begin{aligned} \min \quad & f_0(x) && (4.1) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m && (4.2) \\ & h_i(x) = 0, \quad i = 1, \ldots, p && (4.3) \end{aligned}$$

Let the domain be $\mathcal{D} = (\cap_{i=0}^{m} \mathbf{dom} f_i) \cap (\cap_{i=1}^{m} \mathbf{dom} h_i)$ We introduce variables λ and ν to form the Lagrangian $L(x, \lambda, \nu)$:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \tag{4.4}$$

The variables λ and ν appear to be a weighted sum of functions in the Lagrangian. Their real purpose is to serve as a weighted sum of gradients of the functions, which we see in the KKT conditions.

4.1.2 Lagrange dual function

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right) \tag{4.5}$$

4.1.3 Lower bound on optimal value

$$g(\lambda, \nu) \leq p^\star, \text{ where } \lambda \succeq 0. \tag{4.6}$$

4.2 Lagrange dual problem

$$\begin{aligned} \max \quad & g(\lambda, \nu) && (4.7) \\ \text{subject to} \quad & \lambda \succeq 0 && (4.8) \end{aligned}$$

4.2.1 Weak Duality

Let $d^\star$ be the optimum value of the Lagrange dual problem. Then $d^\star \leq p^\star$ always holds, and this is the best lower bound on $p^\star$, even if the primal problem is not convex. This property is known as *weak duality*

4.2.2 Strong Duality

If the equality

$$d^\star = p^\star \tag{4.9}$$

holds, then we say that *strong duality* holds.

If a problem is convex, we usually (but not always) have strong duality. Additional conditions that establish strong duality are called *constraint qualifications.*

One simple constraint qualification is *Slater's condition:* There exists an $x \in \textbf{relint}\mathcal{D}$ such that

$$f_i(x) < 0, \quad i = 1, \dots, m, \quad Ax = b \tag{4.10}$$

If any inequality constraints are affine, they don't need to hold with strict inequality.

The affine hull of a set is the set of linear combinations of elements of S. The relative interior of a set is the interior relative to the affine hull of the set. For a line connecting two points in $\mathbb{R}^3$, the interior is of course empty, but the relative interior is the open line segment between the points.

4.3 Optimality Conditions

4.3.1 Certificate of suboptimality and stopping criteria

A primal feasible point x and dual feasible pair (λ, ν) establish that

$$p^\star \in [g(\lambda, \nu), f_0(x)] \tag{4.11}$$

The stopping criterion $f_0(x^{(k)}) - g(\lambda^{(k)}, \nu^{(k)}) \leq \epsilon_{abs}$ *guarantees* that the solution is ϵ_{abs}-suboptimal. Alternatively,

$$g(\lambda^{(k)}, \nu^{(k)}), \frac{f_0(x^{(k)}) - g(\lambda^{(k)}, \nu^{(k)})}{g(\lambda^{(k)}, \nu^{(k)})} \leq \epsilon_{rel} \tag{4.12}$$

holds or

$$f_0(x^{(k)}) < 0, \frac{f_0(x^{(k)}) - g(\lambda^{(k)}, \nu^{(k)})}{-f_0(x^{(k)})} \leq \epsilon_{rel}, \tag{4.13}$$

then $p^\star \neq 0$ and the relative error

$$\frac{f_0(x) - p^\star}{|p^\star|} \leq \epsilon_{rel} \tag{4.14}$$

is guaranteed to be less than or equal to ϵ_{rel}

4.3.2 Complementary Slackness

Suppose that strong duality holds. Let $x^\star$ be a primal optimal point and $(\lambda^\star, \nu^\star)$ be a dual optimal point. Then

$$\begin{align}
f_0(x^\star) &= g(\lambda^\star, \nu^\star) \tag{4.15}\\
&= \inf_x \left(f_0(x) + \sum_{i=1}^m \lambda_i^\star f_i(x) + \sum_{i=1}^p \nu_i^\star h_i(x) \right) \tag{4.16}\\
&\leq f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star f_i(x^\star) + \sum_{i=1}^p \nu_i^\star h_i(x^\star) \tag{4.17}\\
&\leq f_0(x^\star) \tag{4.18}
\end{align}$$

We conclude that

$$\lambda_i^\star f_i(x^\star) = 0, \quad i = 1, \dots, m \tag{4.19}$$

4.3.3 KKT Conditions

We assume the functions $f_0, f_1, \ldots, f_m, h_1, \ldots, h_p$ are differentiable.

The idea is that the gradient of the Lagrangian at $x^\star$ must be zero

$$\nabla f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star \nabla h_i(x^\star) = 0 \tag{4.20}$$

Nonconvex problems The constraints, non-negativity of λ_i, complementary slackness, and gradient of Lgrangian together form the KKT conditions:

$$f_i(x^\star) \leq 0, \quad i \in 1, \ldots, m \tag{4.21}$$

$$h_i(x^\star) = 0, \quad i \in 1, \ldots, p \tag{4.22}$$

$$\lambda_i^\star \geq 0, \quad i \in 1, \ldots, m \tag{4.23}$$

$$\lambda_i^\star f_i(x^\star) = 0, \quad i \in 1, \ldots, m \tag{4.24}$$

$$\nabla f_0(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla f_i(x^\star) + \sum_{i=1}^{p} \nu_i^\star \nabla h_i(x^\star) = 0 \tag{4.25}$$

To summarize, for *any* optimization problem with differentiable objective and constraint functions for which strong duality obtains, any pair of primal and dual optimal points must satisfy the KKT conditions. That is, these conditions are necessary

Convex problems. For convex problems, the KKT conditions are also sufficient.

So, if a convex optimization problem satisfies Slater's condition, so that it is strongly dual, then the KKT conditions are necessary and sufficient conditions for optimality.

In effect, we may be able use the KKT conditions to design an algorithm to solve the original convex optimization problem. Often, this algorithm is simpler than the one used to solve the original problem.

Part II

Algorithms

Chapter 5

Unconstrained Minimization

5.1 Unconstrained minimization problems

The general problem is

$$\min f(x) \tag{5.1}$$

where f is twice continuously differentiable and convex. Assume that the problem is solvable: $\exists x^\star$, an optimal point.

A necessary and sufficient condition for a point $x^\star$ to optimal is

$$\nabla f(x^\star) = 0 \tag{5.2}$$

Initial set Let $x^{(0)} \in \mathbf{dom}\ f$

$$S = \{x \in \mathbf{dom}\ f\colon f(x) \leq f(x^{(0)})\} \tag{5.3}$$

5.1.1 Examples

Quadratic minimization and least squares

$$\min \quad \frac{1}{2}x^TPx + q^Tx + r \tag{5.4}$$

The optimality condition (5.2) becomes $Px^\star + q = 0$, which is a set of linear equations.

The least squares problem $\min\|Ax - b\|_2^2$ yields optimality condition

$$A^TAx^\star = A^Tb \tag{5.5}$$

5.1.2 Strong convexity and implications

A function is *strongly convex* on S if there exists an $m > 0$ such that

$$\nabla^2 f(x) \geq mI \tag{5.6}$$

for all $x \in S$. Strong convexity has several interesting conseqeuences.

For one, we get a better lower bound on the function than plain convexity:

$$f(y) = f(x) + \nabla f(x)^T(y - x) + \frac{1}{2}(y - x)^T\nabla^2 f(z)(y - x) \tag{5.7}$$

for some z on the line segment $[x, y]$. Then, for strongly convex functions,

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{m}{2}\|y-x\|_2^2 \tag{5.8}$$

for all x and y in S.

For two, we may relate suboptimality of the current point to the notm of the gradient at the current point. To do so, we calculate the minimizer of the RHS of (5.8) over all y, which is $\tilde{y} = x - \frac{1}{m}\nabla f(x)$. the LHS is larger than that minimum value, so

$$f(y) \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2. \tag{5.9}$$

This expression holds for all y, including the optimal, so that

$$p^\star \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2 \tag{5.10}$$

$$\implies f(x) - p^\star \leq \frac{1}{2m}\|\nabla f(x)\|_2^2 \tag{5.11}$$

Therefore,

$$\|\nabla f(x)\|_2 \leq (2m\epsilon)^{\frac{1}{2}} \implies f(x) - p^\star \leq \epsilon \tag{5.12}$$

Through the Cauchy Shwartz inequality and (5.8) for $y = x^\star$, we can derive

$$\|x - x^\star\|_2 \leq \frac{2}{m}\|\nabla f(x)\|_2^2 \tag{5.13}$$

Upper bound on $\nabla^2 f(x)$ Inequality (5.8) implies that the sublevel sets contained in S are bounded, so that S is bounded. The maximum eigenvalue of $\nabla^2 f(x)$ is bounded on S, since $\nabla^2 f(x)$ is a continuous function. Therefore, there exists a positive constant M such that

$$\nabla^2 f(x) \preceq MI. \tag{5.14}$$

We can then bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{M}{2}\|y-x\|_2^2 \tag{5.15}$$

Analogous to (5.12), by minimizing over y we derive

5.2 Descent Methods

The algorithms in this chapter produce a minimzing sequence $x^{(k)}$, $k = 1, \ldots,$ where

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)} \tag{5.16}$$

and $t^{(k)} > 0$.

All methods are descent methods where $f(x^{(k+1)}) < f(x^{(k)})$ except when $x^{(k)}$ is optimal. The search direction in a descent method must satisfy

$$\nabla f(x^{(k)})^T \Delta x^{(k)} < 0 \tag{5.17}$$

General descent method:

1. Choose a starting point
2. Repeat until stopping criterion is satisfied
   (a) Determine a descent direction
   (b) *Line search.* Choose a step size $t > 0$
   (c) *Update.* $x \leftarrow x + t\Delta x$

Line search is also called *ray search.*

Exact line search Solves a minimization problem along line to determine step size.

$$t = \arg\min_{s\geq 0} f(x + s\Delta x) \tag{5.18}$$

Backtracking line search Depends on α, β with $0 < \alpha < 0.5$, $0 < \beta < 1$. We know that $f(x + t\Delta x) \geq f(x) + t\nabla f(x)^T \Delta x$. We can also bound it locally using the function

$$f(x + t\Delta x) \leq f(x) + \alpha t\nabla f(x)^T \Delta x \tag{5.19}$$

for $0 < \alpha < 1$ Backtracking line search finds the boundary of the region where this bound holds, by starting with $t = 1$ and reducing t by a factor β until

$$f(x + t\Delta x) \leq f(x) + \alpha t\nabla f(x)^T \Delta x. \tag{5.20}$$

Let t_0 satisfy $f(x + t_0\Delta x) = f(x) + \alpha t_0 \nabla f(x)^T \Delta x$. The backtracking stops when $t \leq t_0$, which happens at $t = 1$ if $1 \leq t_0$ or at $t \in (\beta t_0, t_0]$, where the update jumps from $t' > t_0$ to $\beta t' \leq t_0$.

I suspect the reason that we restrict α to $(0, 0.5)$ comes from using the upper bound in (5.15) on $f(x+t\Delta x)$ arising from condition $\nabla^2 f(x) \preceq MI$.

5.3 Gradient Descent Methods

1. Choose a starting point
2. Repeat until stopping criterion is satisfied
   (a) $\Delta x = -\nabla f(x)$
   (b) *Line search.* Choose a step size $t > 0$ via exact or backtracking
   (c) *Update.* $x \leftarrow x + t\Delta x$

Interpretation. As described by Ryan Tibshirani, we may view gradient descent as

$$x^{new} = \arg\min_z f(x^{old}) + \nabla f(x^{old})^T (z - x^{old}) + \frac{1}{2}\|x^{old} - z\|^2,$$

which is a second-order approximation to $f(z)$ except that the Hessian is replaced with the identity function. As we'll see later, in proximal operators, $x^{new} = \mathbf{prox}_{f_{lin}(x)}(x^{old})$, where $f_{lin}(x)$ is the first order approximation to f at x^{old}

Analysis. Let $\tilde{f}(t)$ be $f(x - t\nabla f(x))$. Since f is strongly convex, $mI \preceq \nabla^2 f(x) MI$, we can use the bound in (5.15) to derive

$$\tilde{f}(t) \leq f(x) - t\|\nabla f(x)\|_2^2 + \frac{Mt^2}{2}\|\nabla f(x)\|_2^2 \tag{5.21}$$

If we use exact line search, we would choose t that minimizes the RHS. This choice is $t = 1/M$, and then we have

$$f(x^+) = \tilde{f}(t_{exact}) \leq f(x) - \frac{1}{2M}\|\nabla f(x)\|_2^2 \tag{5.22}$$

$$\implies f(t_{exact}) - p^\star \leq f(x) - p^\star - \frac{1}{2M}\|\nabla f(x)\|_2^2 \tag{5.23}$$

Using the bound (5.11), we get

$$f(t_{exact}) - p^\star \leq \left(1 - \frac{m}{M}\right)(f(x) - p^\star) \tag{5.24}$$

Repeated iterates would yield, with $c = 1 - m/M$,

$$f\left(x^{(k)}\right) - p^\star \leq c^k \left(f\left(x^{(0)}\right) - p^\star\right) \tag{5.25}$$

What this means is that the number of iterations to reach a certain suboptimality ϵ depend on both ϵ and the initial condition. This number is no more than

$$\frac{\log\left(\left(f\left(x^{(0)}\right) - p^\star\right)/\epsilon\right)}{\log\left(1/c\right)} \tag{5.26}$$

The denominator suggest that a high condition number m/M for f will take longer to converge than a small one. Informally, bowls are quick to solve, canyons involve zig-zagging across the path towards the minimum.

Analysis of Backtracking We begin by noting that $\Delta x = -\nabla f(x)$, and that the minimum of the upper bound of $f(x)$ occurs at $t = 1/M$. For $0 \leq t \leq 1/M$, this upper-bound itself is bounded by the linear function $f(x) + \alpha t \nabla f(x)^T \Delta x$, when $\alpha < 0.5$. So,

$$\tilde{f}(t) \leq f(x) - t\alpha \|\nabla f(x)\|_2^2. \tag{5.27}$$

So, backtracking line search on gradient descent will stop either when $t = 1$ or $t \geq \beta t_0 = \beta/M$ Then, the updated value will be

$$f(x^+) \leq f(x) - \alpha\|\nabla f(x)\|_2^2 \text{ or } f(x^+) \leq f(x) - \frac{\beta\alpha}{M}\|\nabla f(x)\|_2^2 \tag{5.28}$$

or simply,

$$f(x^+) \leq f(x) - \min\{\alpha, \beta\alpha/M\}\|\nabla f(x)\|_2^2. \tag{5.29}$$

We're at the same place as the exact line search, where we add $-p^\star$ to both sides. We get that

$$f\left(x^{(k)}\right) - p^\star \leq c^k \left(f\left(x^{(0)}\right) - p^\star\right) \tag{5.30}$$

with

$$c = 1 - \min\{2\alpha m, 2\beta\alpha m/M\} \tag{5.31}$$

Conclusions From the numerical examples shown (in CVXBOOK), and others, we can make the conclusions summarized below.

- The gradient method often exhibits approximately linear convergence, i.e., the error $f(x^{(k)}) - p^\star$ converges to zero approximately as a geometric series.
- The choice of backtracking parameters α, β has a noticeable but not dramatic effect on the convergence. An exact line search sometimes improves the con- vergence of the gradient method, but the effect is not large (and probably not worth the trouble of implementing the exact line search).
- The convergence rate depends greatly on the condition number of the Hessian, or the sublevel sets. Convergence can be very slow, even for problems that are moderately well conditioned (say, with condition number in the 100s). When the condition number is larger (say, 1000 or more) the gradient method is so slow that it is useless in practice.

The main advantage of the gradient method is its simplicity. Its main disadvantage is that its convergence rate depends so critically on the condition number of the Hessian or sublevel sets.

5.4 Steepest Descent

A first-order approximation $f(x+\Delta x) \approx f(x) + \nabla f(x)^T \Delta x$ suggest choosing Δx to make $\nabla f(x)^T \Delta x$ small. Due to the linearity in Δx, we search over a norm-limited region for Δx. A *normalized descent direction* Δx_{nsd} is chosen as

$$\Delta x_{nsd} = \arg \min_{\|v\|=1} \nabla f(x)^T v. \tag{5.32}$$

We consider the *unnormalized* steepest descent direction Δx_{sd} as

$$\Delta x_{sd} = \|\nabla f(x)\|_* \Delta x_{nsd} \tag{5.33}$$

where $\|\cdot\|_*$ denotes the dual norm. Consider γ, $\tilde{\gamma}$ such technicality

$$\|x\| \geq \gamma \|x\|_2, \quad \|x\|_* \geq \tilde{\gamma}\|x\|_2$$

Convergence Analysis. Again assume that f is strongly convex. We derive the same linear convergence rate, with

$$c = 1 - 2m\alpha\tilde{\gamma}^2 \min\{1, \beta\gamma^2/M\}. \tag{5.34}$$

Euclidean norm Steepest descent becomes gradient descent when the norm is the 2-norm: $\Delta x_{sd} = -\nabla f(x)$.

Quadratic norm With norm given by $P \succ 0$, we have $\Delta x_{sd} = -P^{-1}\nabla f(x)$

5.5 Newton's Method

Newton's method is like a steepest descent direction with $P = \nabla^2 f(x)^T$

1. Given a starting point x and tolerance $\epsilon > 0$
2. Repeat until stopping criterion is satisfied
   (a) Compute the Newton step and decrement

   $$\Delta x + nt = -\nabla^2 f^{-1}(x)\nabla f(x); \ \lambda^2 = \nabla f(x)^T \nabla^2 f(x) \nabla f(x)$$

   (b) Stop if $\lambda^2 \leq 2\epsilon$
   (c) *Line search.* Choose a step size $t > 0$ by backtracking line search
   (d) *Update.* $x \leftarrow x + t\Delta x_{nt}$

While gradient descent had linear convergence, Newton's method has quadratic convergence.

5.6 Self-concordance

The convergence rate of Newton's methods involves constants that are hard to estimate. Self-concordant functions allow a convergence analysis that does not depend on unknown constants.

5.7 Sub-Gradient Method

From Ryan Tibshirani

Assume $f(x)$ is convex and $\mathrm{dom} f = \mathbb{R}^n$. The subgradient method is an iteration:

Initialize $x^(0)$ and repeat:

$$x^k \leftarrow x^{k-1} - t_k g^{k-1} \text{ where } g^{k-1} \in \partial f(x^{k-1}).$$

If f is Lipschitz, then subgradient method has a convergence rate $\mathcal{O}(1/\epsilon^2)$, which is considered slow. In their notes, they consider the case of decomposable functions, and show how for such a function we can achieve a convergence rate of $\mathcal{O}(1/\epsilon)$.

The step sizes are either

1. Fixed $t_k = t$, all $k = 1, 2, \ldots$.
2. Diminishing step sizes: choose to meet conditions

$$\sum_{k=1}^{\infty} t_k^2 < \infty, \quad \sum_{k=1}^{\infty} t_k = 0$$

Key difference from gradient descent is pre-specified step size, no adaptation.

5.7.1 First-order Optimality Conditions

$$x^\star \in \arg\min_{x \in \mathcal{D}} f(x) \iff 0 \in \partial f(x^\star) \tag{5.35}$$

Since the subgradient method is not a descent method, it is common to keep track of the best point found so far, i.e., the one with smallest function value. Instead of depending on a strict decrease of the objective function, the sub-gradient method is focused on strictly approaching the optimal set. Helps with constrained optimization

Chapter 6

Equality Constrained Minimization

6.1 Penalty Methods

Consider the problem

$$\begin{aligned} \min \quad & f_0(x) && (6.1)\\ \text{subject to} \quad & h_i(x) = 0, \quad i = 1, \ldots, m && (6.2) \end{aligned}$$

In quadratic penalty methods, this problem is converted into a sequence of unconstrained optimization problems, each of the form

$$\min f_0(x) + \frac{\mu}{2} \sum_{i=1}^{m} h_i(x)^2 \tag{6.3}$$

The sequence corresponds to a sequence $\{\mu^{(k)}\}$ for $k = 0, \ldots, \infty$, where $\mu^{(k)} \to \infty$.

The limit points of the sequence $\{x^{(k)}\}$ corresponding to $\{\mu^{(k)}\}$ converges to an optimum

An issue is that when μ^k is high, the optimization problems may become ill-conditioned.

Non-smooth penalty functions, such as the L_1 norm, may also be used.

6.2 Dual Ascent

Consider

$$\begin{aligned} \min \quad & f(x)\\ \text{subject to} \quad & Ax = b \end{aligned}$$

- Lagrangian: $L(x, y) = f(x) + y^T(Ax - b)$
- Dual function: $g(y) = \inf_x L(x, y)$
- Dual problem: maximize $g(y)$ to get y^*
- Recover $x^* = \arg\min_x L(x, y^*)$

The dual ascent algorithm is one way to use the dual problem to solve the primal problem. A key idea is that

$$\nabla g(y) = Ax^+ - b, \text{ where } x^+ = \arg\min_x L(x, y).$$

The algorithm becomes

$$\begin{aligned} x^{k+1} =& \arg\min_x L(x, y^k)\\ y^{k+1} =& y^k + \alpha^k(Ax^{k+1} - b) \end{aligned}$$

The problem must satisfy some strong assumptions for all this to work. Most problems fail to satisfy these assumptions, and dual ascent cannot be applied. The one case where dual ascent helps is when one can decompose the state into subsystems, with the objective function mirroring this division.

6.3 Dual Decomposition

When the objective is separable ($f(x) = \sum_i^N f_i(x)$), the update of dual ascent may be separated:

$$\begin{aligned} x_i^{k+1} =& \arg\min_{x_i} L_i(x_i, y^k) \\ y^{k+1} =& y^k + \alpha^k(Ax^{k+1} - b) \end{aligned}$$

While the first equation may be performed in parallel, the second requires a *broadcast* and a *gather* operation. Note that we view $Ax = \sum_i^N A_i x_i$.

6.4 Augmented Lagrangians

This method is also known as the *method of multipliers*. It is related to the quadratic penalty algorithm, but it reduces the possibility of ill conditioning by introducing explicit Lagrange multiplier estimates into the function to be minimized, which is known as the augmented Lagrangian function. The augmented Lagrangian adds quadratic penalties to the Lagrangian.

Nocedal and Wright approach AL through quadratic penalty methods with added dual variables, Boyd and Vanderberghe approach it through dual ascent with a quadratic penalty term added.

Define a new lagrangian

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + \frac{\rho}{2}\|Ax - b\|_2^2. \tag{6.4}$$

Applying the dual ascent approach leads to

$$\begin{aligned} x^{k+1} =& \arg\min_{x} L_\rho(x, y^k) \\ y^{k+1} =& y^k + \rho(Ax^{k+1} - b). \end{aligned}$$

which is known as the method of multipliers for solving the original problem. Note that $\alpha^k = \rho$. The method of multipliers converges under far more general conditions than dual ascent, including cases when f takes on the value $+\infty$ or is not strictly convex.

6.5 ADMM

ADMM is an algorithm that is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers. One technicality to overcome is that the square term in the MoM prevents the decomposition in dual ascent. The algorithm solves problems in the form

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{subject to} \quad & Ax + Bz = c \end{aligned}$$

The ADMM algorithm is

$$\begin{aligned} x^{k+1} &= \arg\min_{x} L_\rho(x, z^k, y^k) \\ z^{k+1} &= \arg\min_{z} L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c). \end{aligned}$$

The method of multipliers would have been

$$\begin{aligned} (x^{k+1}, z^{k+1}) &= \arg\min_{x,z} L_\rho(x, z, y^k) \\ y^{k+1} &= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c), \end{aligned}$$

where there is no alternation in solving for x and z. This alternation in ADMM is possible precisely because f and g are separate functions of x and z respectively.

ADMM is most useful when f and g have proximal operators are easy to calculate, but $f + g$ is not so.

Case: $A = I$

When $A = I$, the x-update in ADMM boils down to the proximal operator. When f has structure, then this proximal update is known in closed-form.

Bi-convex problems

ADMM can be applied to biconvex problems

$$\begin{array}{rl} \min & F(x,z) \\ \text{subject to} & G(x,z) = 0 \end{array}$$

The ADMM algorithm is

$$\begin{aligned} x^{k+1} &= \arg\min_x \left(F(x, z^k) + \frac{\rho}{2}\|G(x, z^k) + u^k\| \right) \\ z^{k+1} &= \arg\min_z \left(F(x^{k+1}, z) + \frac{\rho}{2}\|G(x^{k+1}, z) + u^k\| \right) \\ u^{k+1} &= u^k + \rho G(x^{k+1}, z^{k+1}). \end{aligned}$$

6.6 Conjugate Gradient Method

6.6.1 Goal

Solve the equations $Ax = b$, where $A = A^T$ and $A > 0$. This equation arises, for example, from first order optimality conditions involving quadratic objections.

6.6.2 Conjugate Vectors

Two vectors $\mathbf{v}$ and $\mathbf{u}$ are conjugate with respect to A if $\mathbf{v}^T A\mathbf{u} = 0$.

n mutually conjugate vectors with respect to matrix A form a basis for $\mathbb{R}^n$.

Chapter 7

General Constrained Minimization

7.1 Introduction

The goal is to solve general problems of the form

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

Historical note. My early work on classifier-in-the-loop systems formulated training as a projected gradient descent algorithm. Back then I tried to read the blog by Thibaut Lienart to understand PGD, but failed. It was the book on Nonsmooth Optimization and Control Theory by F. Clarke that helped me understand these same terms.

7.2 Proximal Point Algorithm

Algorithm.

$$x^{k+1} \leftarrow \mathbf{prox}_{\lambda f}(x^k)$$

Majorization-minimization. We [1] first interpret the proximal gradient method as an example of a majorization-minimization algorithm, a large class of algorithms that includes the gradient method, Newton's method, and the EM algorithm as special cases; A majorization-minimization algorithm for minimizing a function $\varphi \colon \mathbb{R}^n \to \mathbb{R}$ consists of the iteration

$$x^{k+1} = \arg\min_x \hat{f}(x, x^k),$$

where $\hat{f}(x, x^k)$ is an upper bound for f that is tight at x^k, meaning that $\hat{f}(x, x^k) \geq f(x)$ and $\hat{f}(x, x) = f(x)$. The reason for the name should be clear: such algorithms involve iteratively majorizing (upper bounding) the objective and then minimizing the majorization.

Backward vs Forward. Gradient descent may be viewed as a forward Euler discretization of the gradient flow algorithm (Bach's blog). The proximal point algorithm may then be viewed as the backward discretization approach:

$$\text{Forward Euler Disc.(GDA):} \quad x_{k+1} \leftarrow x_k - \nabla f(x_k) \tag{7.1}$$

$$\text{Backward Euler Disc (PPA):} \quad x_{k+1} \leftarrow x_k - \nabla f(x_{k+1}) \tag{7.2}$$

7.3 Proximal Gradient Descent

From slides by Shenlong Wang. Consider the problem

$$\min_x g(x) + h(x),$$

where g is convex, differentiable, and ∇g is Lipschitz. h is convex, but possible non-differentiable. The non-differentiability of h precludes application of some gradient-based methods. Instead, if $\mathbf{prox}_h$ is easy to implement, we may solve this problem through the proximal gradient algorithm:

$$x^{k+1} = \mathbf{prox}_h \left(x^k - \alpha_k \nabla g(x^k) \right) \tag{7.3}$$

This algorithm needs $\mathcal{O}(1/\varepsilon)$ iterations.

Interpretation

Gradient descent was shown to be similar to a proximal method involving a local gradient-based linear approximation to f. When $f(x) = g(x) + h(x)$, we may repeat this idea with $g(x)$ being the differentiable function, and some rearrangment:

$$\begin{align} x^+ &= \arg\min_z g(x) + \nabla g(x)^T (z - x) + \frac{1}{2\alpha} \|x - z\|^2 + h(x) \tag{7.4} \\ &= \arg\min_z \frac{1}{2\alpha} \|z - (x - \alpha \nabla g(x))\|^2 + h(x) \tag{7.5} \\ &= \mathbf{prox}_{\alpha h} (x - \alpha \nabla g(x)) \tag{7.6} \end{align}$$

Why does this work?

Any minimizer z must satisfy $z = \mathbf{prox}_{\alpha h}(x)$, where $x = z - \alpha_k \nabla g(z)$. Our iterates bounce between z and x (reminds me of Amari's description), and for some $h(x)$, $z = x$.

Accelerated Proximal Gradient Method

This algorithm needs $\mathcal{O}(1/\sqrt{\varepsilon})$ iterations.

$$\begin{align} x^{k+1} &= \mathbf{prox}_h \left(y^k - \alpha_k \nabla g(y^k) \right) \tag{7.7} \\ y^{k+1} &= x^{k+1} + \frac{t-1}{t+2} \left(x^{k+1} - x^k \right) \tag{7.8} \end{align}$$

7.3.1 Iterative soft-thresholding algorithm (ISTA)

The main idea is that the proximal mapping for the l_1 norm is the soft-threshold (dead-zone) map.

7.3.2 Projected Gradient Descent

The FOC conditions lead to an iterative algorithm. The derivation is as follows:

$$\begin{align} & u \in N_C^P(x) \tag{7.9} \\ \implies & \alpha u \in N_C^P(x) \tag{7.10} \\ \implies & x + \alpha u \in x + N_C^P(x) \tag{7.11} \end{align}$$

This observation leads to

$$x = \mathrm{proj}_C (x + \alpha u), \quad \text{for } \alpha \geq 0, u \in N_C^P(x) \tag{7.12}$$

Finally, the minimizer $x^\dagger$ satisfies $0 \in \nabla f(x^\dagger) + N_C^P(x^\dagger) \implies -\nabla f(x^\dagger) \in N_C^P(x^\dagger)$. Therefore,

$$x^\dagger = \mathrm{proj}_C \left(x^\dagger - \alpha \nabla f(x^\dagger) \right), \quad \text{for } \alpha \geq 0 \tag{7.13}$$

A simpler explanation is that PGD corresponds to the proximal gradient algorithm when $h(x) = \mathbb{I}_C(x)$.

Projected Sub-Gradient Descent

When f is non-differentiable, we replace $\nabla f(x)$ with a sub-gradient $g \in \partial f(x)$.

$$x_{t+1} = \operatorname{proj}_C (x_t - \alpha_t g)$$

Approximate Projected Sub-Gradient Descent

This proof reveals that the method still works even when P is not the Euclidean projection operator. All that matters is that P should satisfy $P(u) \in C$, and the inequality

$$d(P(u), z) \leq d(u, z)$$

for all $z \in C$. note that you must project to set, unlike your earlier understanding of approximate

7.4 Mirror Descent

7.4.1 From Dual Spaces

The gradient descent algorithm converts an element of dual vector space $(\mathbb{R}^n)^*$ correctly and uniquely to the vector space $\mathbb{R}^n$ simply by transposition: a row vector (dual space) becomes a column vector (vector space). In this way, the gradient becomes a descent direction. The descent direction in the primal space defines the update step.

When using the Euclidean metric to transform between dual and primal spaces, the update step in the primal space may also be interpreted as a step in the dual space, since $x \in \mathbb{R}^n \mapsto x^T \in (\mathbb{R}^n)^*$. The mirror descent algorithm uses a different transformation based on a mirror function $\Phi\colon \mathbb{R}^n \to \mathbb{R}$. Now, $\nabla\Phi\colon \mathbb{R}^n \to (\mathbb{R}^n)^*$ maps to the dual space, and its inverse turns out to be $\nabla\Phi^*$. We first transform x^k to the dual space, take an update step, and transform back to the primal space:

$$x^{k+1} \leftarrow \nabla\Phi^* \left(\nabla\Phi(x^k) - \alpha_k \nabla f(x^k)\right).$$

An additional Bregman projection step to the feasible region may be required (see writeup by Nicholas Harvey): $x^{k+1} \leftarrow \Pi_C^\Phi(x^{k+1})$.

7.4.2 From PGD

As developed by Thibaut Lienart, we may reinterpret the projected gradient descent algorithm as follows:

$$\begin{aligned} x^{k+1} &= \arg\min_{x\in C} \|(x^k - \alpha_k \nabla f(x^k)) - x\|_2^2 && (7.14)\\ &= \arg\min_{x\in C} \|(x - x^k) + \alpha_k \nabla f(x^k))\|_2^2 && (7.15) \end{aligned}$$

Since the 2-norm is based on the inner product, we may rewrite the norm as

$$\begin{aligned} \|(x^k - \alpha_k \nabla f(x^k)) - x\|_2^2 &= \|x - x^k\|_2^2 + 2\langle x - x^k, \alpha_k \nabla f(x^k)\rangle + \|\alpha_k \nabla f(x^k))\|_2^2 && (7.16)\\ &= \|x - x^k\|_2^2 + 2\alpha_k \langle x, \nabla f(x^k)\rangle + M(x^k) && (7.17) \end{aligned}$$

Now we may rewrite the PGD as

$$\begin{aligned} x^{k+1} &= \arg\min_{x\in C} \|(x^k - \alpha_k \nabla f(x^k)) - x\|_2^2 && (7.18)\\ &= \arg\min_{x\in C} \left\{ \langle x, \nabla f(x^k)\rangle + \frac{1}{\alpha_k} \frac{\|x - x^k\|_2^2}{2} \right\} && (7.19) \end{aligned}$$

Naturally, we may generalize the distance function, leading to *Generalized PGD*, as

$$x^{k+1} = \arg\min_{x\in C} \left\{ \langle x, \nabla f(x^k)\rangle + \frac{1}{\alpha_k} d(x, x^k) \right\} \qquad (7.20)$$

One choice for the distance function d is the Bregman divergence D_φ of some function φ that is μ-strongly convex and differentiable (φ is the **Mirror** function):

$$D_\varphi(x,y) = \varphi(x) - \varphi(y) - \langle x-y, \nabla\varphi(y)\rangle \tag{7.21}$$

In effect, we're trying to minimize $\alpha_k\langle x, \nabla f(x^k)\rangle + \varphi(x) - \langle x, \nabla\varphi(x^k)\rangle + \mathbb{I}_C(x)$. The FOC would then imply that x^{k+1} satisfies

$$0 \in \alpha_k \nabla f(x^k) + \nabla\varphi(x^{k+1}) - \nabla\varphi(x^k) + N_C^P(x^{k+1}) \tag{7.22}$$

$$\implies x^{k+1} = \left(\nabla\varphi + N_C^P\right)^{-1}\left(\nabla\varphi(x^k) - \alpha_k\nabla f(x^k)\right) \tag{7.23}$$

Let $\gamma_o(C)$ denote the set of proper and lsc convex functions on C. It may be shown that for such functions, $(\partial f)^{-1} \equiv \partial f^*$. Therefore, from this page, we define $\phi = \varphi + \mathbb{I}_C$, and then

$$\left(\nabla\varphi + N_C^P\right)^{-1} = \nabla\phi^*,$$

so that the algorithm becomes

$$x^{k+1} = \nabla\phi^*\left(\nabla\varphi(x^k) - \alpha_k\nabla f(x^k)\right) \tag{7.24}$$

Notes:

- The map $\nabla\phi^*$ is the Bregman projection to C.
- If if $N_C^P(x) \subseteq \varphi(x)$, then we should obtain the update $x^{k+1} = \nabla\varphi^*\left(\nabla\varphi(x^k) - \alpha_k\nabla f(x^k)\right)$.
- Principles for choosing φ / D_φ:
  - fits the local curvature of f
  - fits the geometry of the constraint set C
  - makes sure the Bregman projection is inexpensive

7.5 Primal-Dual Methods

From slides by Shenlong Wang.

- Primal: $\min_{x\in X} f(Kx) + g(x)$
- Dual: $\max_{y\in X^*} -f^*(y) - g^*(-K^*y)$
- Primal-dual: $\min_{x\in X}\max_{y\in X^*} \langle Kx, y\rangle + g(x) - f^*(y)$

The primal-dual form is useful when $\mathbf{prox}_f$ is difficult, but $\mathbf{prox}_{f^*}$ and $\mathbf{prox}_g$ are easier.

The saddle point $\hat{x}, \hat{y}$ should satisfy

$$0 \in K\hat{x} - \partial f^*(\hat{y}) \tag{7.25}$$

$$0 \in K^*\hat{y} + \partial g(\hat{x}) \tag{7.26}$$

These conditions lead to an iterative algorithm, since they are telling us what the sub-gradients are. Choose step size σ, θ and τ such that $\sigma\tau L^2 < 1$, where $L = |K\|$, $\theta \in [0,1]$:

$$y^{k+1} = \mathbf{prox}_{f*}\left(y^k + \sigma K x^k\right) \tag{7.27}$$

$$\bar{x}^{k+1} = \mathbf{prox}_g\left(\bar{x}^k - \tau K^* y^{k+1}\right) \tag{7.28}$$

$$x^{k+1} = \bar{x}^k + \theta\left(\bar{x}^{k+1} - \bar{x}^k\right) \tag{7.29}$$

Primal-dual method is equivalent to ADMM if $K = I$. But in the general case primal-dual is usually faster, since solving the subproblems of ADMM is harder.

7.6 Proximal Flows

Gradient flow. Proximal minimization can be interpreted as a discretized method for solving a differential equation whose equilibrium points are the minimizers of a differentiable convex function f. The differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = -\nabla f(x(t)) \tag{7.30}$$

is called the gradient flow of f. The equilibrium points of the gradient flow are the zeros of ∇f, the minimizers of f.

Subgradient differential Inclusion The idea of the gradient flow can be generalized to cases where f is non-differentiable via the *subgradient differential inclusion*

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) \in -\partial f(x(t)) \tag{7.31}$$

Bibliography

[1] Parikh, N., & Boyd, S. (2014). Proximal algorithms. Foundations and Trends in optimization, 1(3), 127-239.

[2] Clarke, F. H., Ledyaev, Y. S., Stern, R. J., & Wolenski, P. R. (2008). Nonsmooth analysis and control theory (Vol. 178). Springer Science & Business Media.