Lecture notes on Optimization for MDI210

Robert M. Gower

September 30, 2019

Abstract

Theses are my notes for my lectures for the MDI210 Optimization and Numerical Analysis course. Theses notes are a work in progress, and will probably contain several mistakes (let me know?). If you are following my lectures you may find them useful to recall what we covered in class. Otherwise, I recommend you read these lectures notes [1] for the part on linear programming the excellent book [3] for the nonlinear optimization part. In particular, this book [3] contains all the subjects covered in these notes, and is a much better reference than these notes.

Contents

1 Linear Programming

Consider the problem

$$\begin{aligned} \max_x \, z &\overset{\text{def}}{=} c^\top x \\ \text{subject to } Ax &\leq b, \\ x &\geq 0, \end{aligned} \tag{LP}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. We assume that $m \leq n$, otherwise the feasible set might be trivial. For instance if $n \leq m$ and full rank, then there exists only one feasible point or no feasible points.

1.1 A first example

First we start with a simple 2D graphic example. We want to maximize the production of products x and y. Each unit of product x and y gives us 4 and 2 profit, respectively. To produce one unit of product x we need 30 minutes on machine 1 and 40 minutes on machine 2. To produce one unit of product y we need 20 minutes on machine 1 and 10 minutes on machine 2. Thus our problem is to maximize $4x + 2y$ subject to.

$$\begin{aligned} 3x + 2y &\leq 6 \\ 4x + 1y &\leq 4 \end{aligned}$$

See Figures 1 for a graphical illustration. This graphical example leads us to believe that is the solution is attainable, then it is always at a vertex. Indeed, this is the case as we show in Theorem 1.1.

1.2 Fundamental Theorem of linear programming

Theorem 1.1 (Fundamental Theorem of Linear Programming)**.** Let $P = \{x \,|\, Ax = b, x \geq 0\}$. One of the three circumstances must hold

1. $P = \{\emptyset\}$

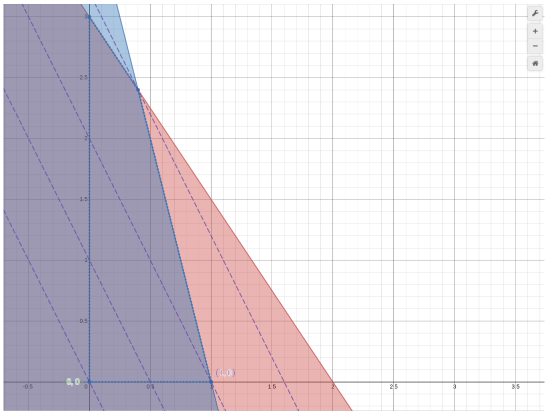


Figure 1: An graphical example of the constraint set together with the level sets of $4x + 2y = z$ with $z \in \{0, 2, 4, 6.4\}$ as dashed lines.

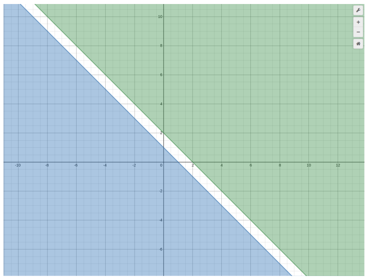
2. $P \neq \{\emptyset\}$ and there exists a vertex v of P such that $v \in \arg\min_{x \in P} c^\top x$

3. There exists $x, d \in \mathbb{R}^n$ such that $x + td \in P$ for all $t \geq 0$ and $\lim_{t \to \infty} c^\top(x + td) = \infty$.

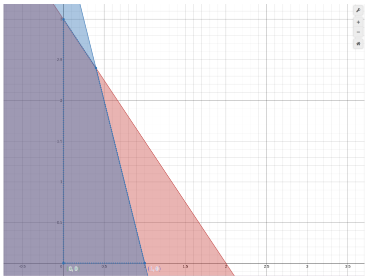
Proof: A formal proof can be found in Section 2.1.2 [1]. Here we will illustrate with examples.

1. The case $P = \{\emptyset\}$ is evidenced with the simple example

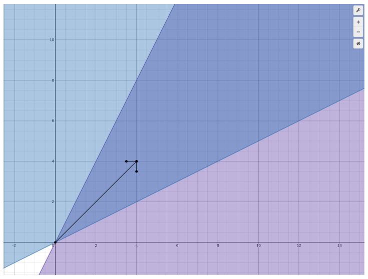
$$\begin{aligned} x + y &\leq 1 \\ x + y &\geq 2 \end{aligned}$$



2. $P \neq \{\emptyset\}$ and there exists a vertex v of P such that $v \in \arg\min_{x \in P} c^\top x$



3. There exists $x, d \in \mathbb{R}^n$ such that $x + td \in P$ for all $t \geq 0$ and $\lim_{t\to\infty} c^\top(x + td) = \infty$.



$$\begin{array}{lrcl} \max & x + y & & \\ & 2x - y & \geq & 0 \\ & x - 2y & \leq & 2 \end{array}$$

This theorem suggests how we should develop an algorithm to solve the LP. First we should take care of finding a feasible point and assuring that the domain is non-empty. This is called **initialization phase** or **phase 1**. After determining a feasible point, we should traverse the vertices of P searching for the optimal point. If we find one, we say that the solution is **attainable**. If we find a direction in which $c^\top x \to \infty$, we say that there is an **unbounded** solution.

Let us transform this insight into a method for solving. First we transform a problem given in the standard form

$$\begin{array}{lll} \max & 4x_1 + 2x_2 & \\ & 3x_1 + 2x_2 & \leq 600 \\ & 4x_1 + 1x_2 & \leq 400 \\ & x_1 \geq 0, x_2 \geq 0. & \end{array} \tag{1}$$

into a maximization over linear constraints with non-negativity constraints

$$\begin{array}{rcccc} \max & 4x_1 & + & 2x_2 & \\ x_3 = 600 & - & 3x_1 & - & 2x_2 \\ x_4 = 400 & - & 4x_1 & - & x_2. \end{array} \tag{2}$$

We arrived at the above (1) by simply adding x_3 and x_4 on the 1st and 2nd rows to fill in the *slack* of the inequality. Accordingly, the variables x_3 and x_4 are referred to as the slack variables.) It is not hard to show that (1) and (2) are equivalent.

Exercise 1.2. Show that if (x_1^*, x_2^*) is a solution to (1) then there exists x_3^* and x_4* such that $(x_1^*, x_2^*, x_3*, x_4*)$ is a solution to (2). Furthermore, if $(x_1^*, x_2^*, x_3*, x_4*)$ is a solution to (2), show that (x_1^*, x_2^*) is a solution to (1).

Proof: Straightforward.

The transformation (1) into (2) has effectively turned the constraint $Ax \leq b$ into a linear system $Ax - b = I\hat{x}$ or equivalently $Ax - I\hat{x} = b$. So now we can apply row transformations (left multiplying by an invertible matrix). Next we re-write (2) in the *Dictionary* format

$$\begin{array}{ccccccc} x_3 & = & 600 & - & 3x_1 & - & 2x_2 \\ x_4 & = & 400 & - & 4x_1 & - & x_2 \\ \hline z & = & & & 4x_1 & + & 2x_2 \end{array}$$

The non-negativity constraints are non longer explicitly included, so we must take care so as to ensure they hold. Note that the above has a convenient *feasible point*, that is, the point $(x_1^*, x_2^*, x_3^*, x_4^*) = (0, 0, 600, 400)$ satisfies the above equality constraints. The feasible point corresponds to the $(0, 0)$ point in our original 2D constraint space. We will now try to move from this vertex to a neighbouring vertex that has a larger objective value.

Note that the *cost coefficient* of x_1 in the objective function is positive, which shows that increasing x_1 will increase the objective function. But increasing x_1 decreases x_3 and x_4. Indeed we have

$$\begin{array}{ccccc} x_3 \geq 0 & \Rightarrow & 600 - 3x_1 \geq 0 & \Rightarrow & x_1 \leq 200, \\ x_4 \geq 0 & \Rightarrow & 400 - 4x_1 \geq 0 & \Rightarrow & x_1 \leq 100. \end{array}$$

Thus $x_1 \leq 100$ otherwise x_4 will become negative. We will now perform row operations on (3) so that x_1 appears isolated on the left-hand side taking x_4's position. This is called *pivoting* on the element $(4, 1)$. This gives

$$\begin{array}{ccccccc} x_3 & = & 300 & & 0 & - & \frac{5}{4}x_2 \\ x_1 & = & 100 & - & \frac{x_4}{4} & - & \frac{x_2}{4} \\ \hline z & = & 400 & - & x_4 & + & x_2 \end{array}$$

Now we are at the vertex $(x_1^*, x_2^*) = (100, 0)$. We refer to this last operation as x_4 *leaving the basis* and x_1 *entering the basis*. Next we see that increasing x_2 increases the objective value but

$$\begin{array}{ccc} x_3 \geq 0 & \Rightarrow & 240 \geq x_2, \\ x_1 \geq 0 & \Rightarrow & 400 \geq x_4. \end{array}$$

Consequently we can increase x_2 upto 240 while respecting the positivity constraints of x_3. Thus x_3 will leave the basis and x_2 will enter the basis. Performing a row elimination again, we have that

$$\begin{array}{ccccccc} x_2 & = & 240 & & 0 & - & \frac{4}{5}x_3 \\ x_1 & = & 40 & - & \frac{x_4}{4} & - & \frac{1}{5}x_3 \\ \hline z & = & 640 & - & x_4 & - & \frac{4}{5}x_3 \end{array}$$

Now increasing x_4 or x_3 will decrease the objective value, thus we can make no further improvement. The final optimal vertex is given by $(x_1^*, x_2^*) = (40, 240)$ and the optimal objective value is $z^* = 640$.

1.3 Notation and definitions

Before formalizing a method for solving the above, we will establish a standard representation of linear programs. The standard form is given in (LP). Not all LPs fit the format (LP), but all LPs can be re-written in the form (LP). A few of the standard tricks we use to re-write an LP in the standard form are listed as here

1. (**Equality**) Replace $\sum_{j=1}^{n} a_{ij}x_j = b_i$ by

   $$\sum_{j=1}^{n} a_{ij}x_j \leq b_i,$$

   and

   $$-\sum_{j=1}^{n} a_{ij}x_j \leq -b_i.$$

2. (**Box constraints**) Replace

   $$\alpha \leq x_i \leq \beta,$$

   by

   $$y = x - \alpha,$$

   $$y \leq \beta - \alpha$$

   and

   $$y \geq 0.$$

3. (**Unconstrained**) If x has no positivity constraint, then replace $x = x^+ - x^-$ and add the constraints $x^+ \geq 0$ and $x^- \geq 0$.

We will now formalize the definitions we introduced in the examples.

- The objective is to maximize the linear objective function $z = \sum_{j=1}^{n} c_j x_j$
- There are m inequality constraints in the standard form given by

  $$\sum_{j=1}^{n} a_{ij}x_j \leq b_i, \text{ for } i \in \{1, \ldots, m\}.$$

- There are n positivity constraints given by $x_j \geq 0$, for $j \in \{1, \ldots, n\}$.
- We call $(x_1^*, \ldots, x_n^*) \in \mathbb{R}^n$ a feasible solution if it satisfies the inequality and positivity constraints.

When passing from the standard form to the dictionary form, we had to introduce some additional notation

- We call the additionally introduced variable $(x_{n+1}, \dots, x_{n+m}) \in \mathbb{R}^m$ the slack variables ("variables d'écart ")
- We refer to

  $$\begin{array}{lcl} x_{n+1} & = & b_1 - \sum_{j=1}^n a_{1j} x_j \\ \vdots & & \\ x_{n+i} & = & b_i - \sum_{j=1}^n a_{ij} x_j \\ \vdots & & \\ x_{n+m} & = & b_m - \sum_{j=1}^n a_{mj} x_j \\ \hline z & = & \sum_{j=1}^n c_j x_j, \end{array}$$

  as the initial dictionary.
- We say that the system of equations with non-negativity constraints forms a valid dictionary if m of the variables $(x_1, \dots, x_{n+m})$ can be expressed as an explicit function of the remaining n variables.
- We refer to the m variables isolated on the left-hand side as the *basic variable* (*variable de base*)and the remaining n variables as *non-basic* (*variable hors-base*) or *outside* the basis.

After writing the problem in the dictionary format, we then performed several row elimination operations to change the basic variables. These row operations altered the coefficients of the constraints a_{ij} and the cost vector c_j. We introduce notation to accommodate for these changing coefficients.

- Let $I \subset \{1, \dots, n+m\}$ be the set of indices of the basic variables and let $J = \{1, \dots, n+m\} \backslash I$ be the non-basic variables.
- For a given basis determined by I there is a corresponding dictionary

  $$\begin{array}{lcl} x_i & = & b'_i + \sum_{j \in J} a'_{ij} x_j, \text{ for } i \in I \\ \hline z & = & z^* + \sum_{j \in J} c'_j x_j, \end{array}$$

  where $a'_{ij}, b'_i, z^* \in \mathbb{R}$ are coefficients resulting from the row operations. For this to be a feasible dictionary we require that $b'_i \geq 0$.

1.4 The simplex algorithm

Let us now formalize the operations of the simplex algorithm. First, we choose a variable x_{j_0} that has a positive cost c'_{j_0} to enter the basis, see Algorithm 1. Otherwise if all costs are negative, we have found the optimal. Next we must choose a variable x_{i_0} that will leave the basis. We determine

i_0 as the first variable whose value equals zero as we increase $x^*_{j_0}$. That is, since $x^*_{j_0}$ will be the only non-zero non-basic variable, we have

$$x^*_i = b'_i + \sum_{j \in J} a'_{ij} x^*_j = b'_i + a'_{ij_0} x^*_{j_0}.$$

Since we require $x^*_i \geq 0$ this imposes that $b'_i + a'_{ij_0} x^*_{j_0} \geq 0$. In other words

$$x^*_{j_0} \leq -\frac{b'_i}{a'_{ij_0}} \quad \text{if } a'_{ij_0} < 0,, \qquad \text{for } i \in I, \tag{3}$$

$$x^*_{j_0} \geq -\frac{b'_i}{a'_{ij_0}} \quad \text{if } a'_{ij_0} \geq 0, \qquad \text{for } i \in I. \tag{4}$$

Since we are only interested in increasing the value of $x^*_{j_0}$, the case (4) where $a'_{ij_0} \geq 0$ does not restrict $x^*_{j_0}$ from increasing (since $b'_i \geq 0$). Thus only (3) constrains the value $x^*_{j_0}$. With the preceding definitions, we can now state the Simplex method in Algorithm 1. In particular, the pivoting step in Algorithm 1 can be stated using elementwise operations as

```
for $j \in J$ do
    for $i \in I \setminus \{i_0\}$ do
        $a'_{ij} \leftarrow a'_{ij} - \frac{a'_{ij_0}}{a'_{i_0 j_0}} a'_{i_0 j}$        # Row elimination on pivot $(i_0, j_0)$.
    $c_j \leftarrow c_j - \frac{c'_{j_0}}{a'_{i_0 j_0}} a'_{i_0 j}$
for $j \in J$ do
    $a_{i_0 j} \leftarrow -\frac{a_{i_0 j}}{a'_{i_0 j_0}}$
$a'_{i_0 j_0} = 1/a'_{i_0 j_0}$
```

What was left unclear is how do we choose j_0 to enter the basis. There are four common choices.

1. The mad hatter rule: Choose the first one you see.

2. Dantzig's 1st rule: $j_0 = \arg\max_{j \in J} c_j$.

3. Dantzig's 2nd rule: Choose $j_0 \in \{j \in J \, : \, c_j > 0\}$ that results in the largest increase in the objective value. Let $t = \min_{i \in I, a_{ij_0} < 0} \left\{ -\frac{b_i}{a_{ij_0}} \right\}$. The variable entering the basis will have value $x^*_{j_0} = t$. Consequently the objective value increases by tc_{j_0}. Choosing j_0 that maximizes this increase is equivalent to choosing via

$$j_0 = \arg\max_{j \in J} \left\{ c_j \min_{i \in I, a_{ij} < 0} \left\{ -\frac{b_i}{a_{ij}} \right\} \right\}.$$

   This effective but computationally expensive.

4. Bland's rule: Choose the smallest indices j_0 and i_0. That is, choose

$$j_0 = \arg\min\{j \in J \, : \, c_j > 0\}.$$

Algorithm 1 One Simplex iteration

```
Input: A basic index set $I \subset \{1, \ldots, n+m\}$, $J = I \setminus \{1, \ldots, n+m\}$, constraint coefficients $a'_{ij} \in \mathbb{R}$, $b'_i \geq 0$ and $c'_i \in \mathbb{R}$.

if $c_i \leq 0$ for all $i \in J$ then
    STOP;        # Optimal point found.
Choose a variable $j_0$ to enter the basis from the set $j_0 \in \{j \in J \,:\, c'_j > 0\}$.
if $a'_{ij_0} \geq 0$ for all $i \in I$ then
    STOP;        # The problem is unbounded.
Choose a variable $i_0$ to leave the basis from the set $i_0 \in \arg \min_{i \in I, a'_{ij_0} < 0} \left\{ -\frac{b'_i}{a'_{ij_0}} \right\}$.
$aux \leftarrow a'_{i_0 j_0}$
for $i \in I \setminus \{i_0\}$ do
    $a'_{i:} \leftarrow a'_{i:} - a'_{ij_0}\, a'_{i_0:} / \, a_{i_0 j_0}$        # Row elimination on pivot $(i_0, j_0)$.
$c' \leftarrow c' - c'_{j_0}\, a'_{i_0:} / \, a_{i_0 j_0}$        # Update the cost coefficients.
$a'_{i_0:} \leftarrow -\, a'_{i_0:} / \, a'_{i_0 j_0}$        # Row normalization .
$a_{i_0 j_0} \leftarrow 1/\, aux$
$I \leftarrow (I \setminus \{i_0\}) \cup \{j_0\}$.        # Update basic variables set
$J \leftarrow (J \setminus \{j_0\}) \cup \{i_0\}$.        # Update non-basic variable set

Output: $I, a'_{ij}, b'_j, c'_j$.
```

If the set $\arg\min_{i \in I, a_{ij_0} < 0} \left\{ \frac{b_i}{a_{ij_0}} \right\}$ has more than one element, choose the smallest

$$i_0 = \min \left\{ \arg \min_{i \in I, a_{ij_0} < 0} \left\{ -\frac{b_i}{a_{ij_0}} \right\} \right\}.$$

Dantzig's rules were designed to maximize the objective function in a greedy manner. While Bland's rule, though apparently mundane, was designed to avoid *cycling*.

1.5 Degeneracy and Cycling

If any of the basic variables have zero value, we say that it is a *degenerate* basis. Degenerate basis require extra care because they may lead to the simplex algorithm cycling. See example on board and your alternative french notes.

1.6 Initialization using a first phase problem

Not always will we have that $b_i \geq 0$ for $i = 1, \ldots, m$. Simply including slack variables will not lead to a feasible basic solution. To find a feasible solution we will use the simplex method on an auxiliary *first phase* problem.

First assume we are given a problem

$$
\begin{aligned}
\max_x z &\overset{\text{def}}{=} c^\top x \\
\text{subject to } Ax &\leq b, \\
x &\geq 0,
\end{aligned}
\tag{5}
$$

where at least one $b_i < 0$ where $i \in \{1, \ldots, m\}$. Consequently $x^* = 0$ is not a feasible solution. To remedy this we add an additional variable x_0 and **change the objective**

$$
\begin{aligned}
\max_x \; &-x_0 \\
\text{subject to } Ax &\leq b + Ix_0, \\
x &\geq 0, x_0 \geq 0.
\end{aligned}
\tag{LP-1st}
$$

There now exists x_0 for which there is a feasible solution, for instance choosing $x_0^* = \max_{i=1,\ldots,m} |b_i|$ and $x_i^* = 0$ for $i = 1, \ldots n$. The problem (LP-1st) is known as the *1st phase* simplex method. We can use the simplex method to solve (LP-1st). If the solution to (LP-1st) is such that $x_0^* \neq 0$, then we know that the original problem (5) is infeasible. If the solution to (LP-1st) is such that $x_0^* = 0$, then we can use the remaining variables $x_i^* \neq 0$ as a starting basis.

To do this, first we setup a dictionary with slack variables as the basis, even though they do not form a feasible basis.

$$
\begin{array}{ccccccc}
x_{n+1} & = & b_1 & - & \sum_{j=1}^n a_{1j}x_j & + & x_0 \\
\vdots & & & & & & \\
x_{n+m} & = & b_m & - & \sum_{j=1}^n a_{mj}x_j & + & x_0 \\
\hline
z & = & & & & - & x_0
\end{array}
\tag{6}
$$

Next, different from our previous examples, we will make x_0 enter the basis even though it has a negative cost. Suppose w.l.o.g that x_{n+1} leaves the basis as x_0 enters. Thus after pivoting on row 1, column $n+1$ and performing the row operations

$$
r_i \to r_i - r_1 \text{ for the } i = 2, \ldots, m,
$$

the next dictionary would be

$$
\begin{array}{ccccccc}
x_0 & = & -b_1 & + & \sum_{j=1}^n a_{1j}x_j & + & x_{n+1} \\
\vdots & & & & & & \\
x_{n+m} & = & b_m - b_1 & - & \sum_{j=1}^n (a_{mj} - a_{1j})x_j & + & x_{n+1} \\
\hline
z & = & b_1 & - & \sum_{j=1}^n a_{1j}x_j & - & x_{n+1}
\end{array}
\tag{7}
$$

Now, so long as there are positive elements in $(a_{1j})_j$, we can proceed with the simplex method as usual. Note that this indicates we should have chosen an element i_0 to leave basis such that $(a_{i_0j})_j$. has many positive coefficients. If we continue to iterate and x_0 leaves the basis, then we have found a feasible basis point and we can drop the x_0 variable.

1.7 Duality

Consider again the LP in standard form

$$\begin{aligned} \max_x \; & z \overset{\text{def}}{=} c^\top x \\ \text{subject to } & Ax \leq b, \\ & x \geq 0, \end{aligned} \tag{P}$$

Though we now have a technique for solving (P), at any given moment we do not know how far we are from the solution. Will we need another few minutes of computing resources or days of computing resources? This is a troublesome question. For this, and other reasons, we will now develop an alternative and equivalent formulation of (P) that will help determine if we are near the solution, among other insights. This equivalent formulation is known as the *dual* formulation. We can first derive the dual problem as a means of finding an upper bound to the solution of (P).

Say we wish to upper bound our objective $z = c^\top x$ and we want to do this by combining rows of the constraints $Ax \leq b$. That is, let $y \geq 0 \in \mathbb{R}^m$. If we could determine such a y so that $y^\top A \approx c^\top$ then we would have that

$$c^\top x \approx (y^\top A)x \leq y^\top b,$$

thus $y^\top b$ is an approximate upper bound of $c^\top x$. What is more, $y^\top b$ does not depend on x, ergo this bound holds for all x, including the optimal solution. But just being approximate upper bound is no good. Instead, assume we have $0 \leq y \in \mathbb{R}^m$ such that $y^\top A \geq c^\top$ or equivalently $A^\top y \geq c$. Then indeed the upper bound holds since

$$c^\top x \leq (y^\top A)x \leq y^\top b.$$

Now say we want the upper bound to be as tight as possible. We can do this by choosing $y \geq 0$ so that $y^\top b$ is small as possible. That is, we need to the following *dual* linear program.

$$\begin{aligned} \min_y \; & w \overset{\text{def}}{=} y^\top b \\ \text{subject to } & A^\top y \geq c, \\ & y \geq 0. \end{aligned} \tag{D}$$

Exercise 1.3. Show that the dual of the dual is the primal program. In other words, this is a *reflexive* transformation.

By construction of the dual program, the following lemma holds.

Lemma 1.4 (Weak Duality)**.** If $x \in \mathbb{R}^n$ is a feasible point for (P) and $y \in \mathbb{R}^m$ is a feasible point for (D) then

$$c^\top x \leq y^\top Ax \leq y^\top b. \tag{8}$$

Consequently

- If (P) has an unbounded solution, that is $c^\top x \to \infty$, then there exists no feasible point y for (D)
- If (D) has an unbounded solution, that is $y^\top b \to -\infty$, then there exists no feasible point x for (P)
- If x and y are primal and dual feasible, respectively, and $c^\top x = y^\top b$, then x and y are the primal and dual optimal points, respectively.

What is even more remarkable is that, not only does (D) provide an upper bound for (P), but they are equivalent problems, in the following sense

Theorem 1.5 (Strong Duality)**.** If (P) or (D) is feasible, then $z^* = w^*$. Moreover, if c' is the cost vector of the optimal dictionary of the primal problem, that is, if

$$z = z^* + \sum_{i=1}^{n+m} c'_i x_i, \tag{9}$$

then $y^*_i = -c'_{n+i}$ for $i = 1, \ldots, m$.

Proof: First note that $c'_i \leq 0$ for $i = 1, \ldots, m+n$ otherwise the dictionary would not be optimal. Consequently $y^*_i = -c'_{n+i} \geq 0$ for $i = 1, \ldots, m$. Furthermore, by the definition of the slack variables we have that

$$x_{n+i} = b_i - \sum_{j=1}^n a_{ij}x_j, \quad \text{for } i = 1, \ldots, m. \tag{10}$$

Consequently, setting $y^*_i = -c'_{n+i}$, we have that

$$\begin{aligned} z &\overset{(9)}{=} z^* + \sum_{j=1}^n c'_j x_j + \sum_{i=n+1}^{n+m} c'_i x_i \\ &\overset{(10)}{=} z^* + \sum_{j=1}^n c'_j x_j - \sum_{i=1}^m y^*_i (b_i - \sum_{j=1}^n a_{ij}x_j) \\ &= z^* - \sum_{i=1}^m y^*_i b_i + \sum_{j=1}^n \left(c'_j + \sum_{i=1}^m y^*_i a_{ij} \right) x_j \\ &= \sum_{j=1}^n c_j x_j, \qquad \forall x_1, \ldots, x_n. \end{aligned} \tag{11}$$

where the last line followed by definition of the objective function $z = \sum_{j=1}^{n} c_j x_j$. Since the above holds for all $x \in \mathbb{R}^n$, we can match the coefficients to obtain

$$\begin{aligned} z^* &= \sum_{i=1}^{m} y_i^* b_i & (12) \\ c_j &= c_j' + \sum_{i=1}^{m} y_i^* a_{ij}, \qquad \text{for } j = 1, \ldots, n. & (13) \end{aligned}$$

Since $c_j' \leq 0$ for $j = 1, \ldots, n$, the above is equivalent to

$$\begin{aligned} z^* &= \sum_{i=1}^{m} y_i^* b_i & (14) \\ \sum_{i=1}^{m} y_i^* a_{ij} &\leq c_j, \qquad \text{for } j = 1, \ldots, n. & (15) \end{aligned}$$

The inequalities (15) prove that y_i^*'s satisfies the constraints in (D), and thus is feasible. The equality (14) shows that $z^* = \sum_{i=1}^{m} y_i^* b_i = w$, and consequently by weak duality the y_i^*'s are dual optimal. □

Calculating the dual optimal variables y^* using Theorem 1.5 requires knowing the cost vector c' of the optimal tableau. But it turns out that we do not need c' to calculate y^*. We can instead recover the y^* by only knowing x^*, and the complementary slackness theorem shows (see notes).

1.8 How to compute dual solution: Complementary slackness

Let $x^* \in \mathbb{R}^n$ be an optimal solution of (P). Then $y^* \in \mathbb{R}^m$ is an optimal dual solution if $c^\top x^* = (y^*)^\top b$. Thus by the weak duality theorem we have that

$$c^\top x^* \quad = \quad (y^*)^\top A x^* \quad = \quad (y^*)^\top b.$$

Subtracting $(y^*)^\top A x^*$ from all sides of the above gives

$$\big(\underbrace{c - A^\top y^*}_{\geq 0}\big)^\top x^* \quad = \quad 0 \quad = \quad (y^*)^\top \big(\underbrace{b - A x^*}_{\geq 0}\big).$$

Re-writing the above in element form we have that

$$\sum_{j=1}^{n} \big(c_j - \sum_{i=1}^{m} a_{ij} y_i^*\big) x_j^* \quad = \quad 0 \quad = \quad \sum_{i=1}^{m} y_i^* \big(b_i - \sum_{j=1}^{n} a_{ij} x_j^*\big).$$

On both sides we have a sum over the product of positive numbers. Thus the total sum is only zero if the individual products are zero, that is

$$\begin{aligned} y_i^* \big(b_i - \sum_{j=1}^{n} a_{ij} x_j^*\big) &= 0, \quad \forall i = 1, \ldots, m. \\ x_j^* \big(c_j - \sum_{i=1}^{m} a_{ij} y_i^*\big) &= 0, \quad \forall j = 1, \ldots, n. \end{aligned}$$

This gives the following rule for computing y^*.

$$\begin{aligned} \sum_{i=1}^{n} a_{ij} y_i^* &= c_j, \quad \forall j \in \{1, \ldots, n\}, \;\; x_j^* > 0. \\ y_i &= 0, \quad \forall i \in \{1, \ldots, m\}, \;\; b_i > \sum_{j=1}^{n} a_{ij} x_j^*. \end{aligned}$$

Thus we need a single system solve to compute y^*. Finally since $b_i > \sum_{j=1}^{n} a_{ij} x_j^*$ implies that the slack variable $x_{n+i}^* > 0$ we have the more succinct rule

$$\begin{aligned} \sum_{i=1}^{n} a_{ij} y_i^* &= c_j, \quad \forall j \in \{1, \ldots, n\}, \;\; x_j^* > 0. \\ y_i &= 0, \quad \forall i \in \{1, \ldots, m\}, \;\; x_{n+i}^* > 0. \end{aligned} \tag{16}$$

Definition 1.6. We refer to the method for computing the dual variables using (16) as the primal dual map. Indeed for any primal feasible values x^* we can compute a dual feasible y^* using (16).

2 Nonlinear programming without constraints

Robert: I recommend reading Chapter 1 in [2] as an introduction into nonlinear optimization.

We now move onto nonlinear optimization, that is, we wish to minimize a possibly nonlinear differentiable function $f : x \in \mathbb{R}^n \mapsto f(x) \in \mathbb{R}$. First we will consider the unconstrained optimization problem

$$x^* \in \arg\min_{x \in \mathbb{R}^n} f(x). \tag{17}$$

All the methods we develop are iterative, in that they produce a sequence of iterates $x^1, \ldots, x^k, \ldots,$ in the hope that they converge to the solution with

$$\lim_{k \to \infty} x^k = x^*.$$

Furthermore, we will focus on *descent methods* where the iterates are calculated via

$$x^{k+1} = x^k + s_k d^k, \tag{18}$$

where $s_k > 0$ is a *step size* and $d^k \in \mathbb{R}^n$ is *search direction.* The search direction and step size will satisfy the *descent condition* given by

$$f(x^k) < f(x^{k+1}). \tag{19}$$

This in turn guarantees that, little by little, we get closer to the solution of (17).

The key tool in ensuring that the descent condition holds is the gradient.

2.1 The gradient, Hessian and the Taylor expansion

For a continuously differentiable function $f : x \in \mathbb{R}^n \mapsto f(x) \in \mathbb{R}$, we refer to $\nabla f(x)$ as the gradient evaluated at x defined by

$$\nabla f(x) \quad = \quad \left[\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}\right]^\top.$$

Note that $\nabla f(x)$ is a column-vector.

For any vector valued function $F : x \in \mathbb{R}^d \to F(x) \quad = \quad [f_1(x), \ldots, f_n(x)]^\top \in \mathbb{R}^n$ we define the *Jacobian matrix* by

$$\nabla F(x) \overset{\text{def}}{=} \begin{bmatrix} \frac{\partial f_1(z)}{\partial x_1} & \frac{\partial f_1(z)}{\partial x_2} & \frac{\partial f_1(z)}{\partial x_3} & \cdots & \frac{\partial f_1(z)}{\partial x_d} \\ \frac{\partial f_2(z)}{\partial x_1} & \frac{\partial f_2(z)}{\partial x_2} & \frac{\partial f_2(z)}{\partial x_3} & \cdots & \frac{\partial f_2(z)}{\partial x_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(z)}{\partial x_1} & \frac{\partial f_n(z)}{\partial x_2} & \frac{\partial f_n(z)}{\partial x_3} & \cdots & \frac{\partial f_n(z)}{\partial x_d} \end{bmatrix} = \begin{bmatrix} \nabla f_1(x)^\top \\ \nabla f_2(x)^\top \\ \nabla f_3(x)^\top \\ \vdots \\ \nabla f_n(x)^\top \end{bmatrix}.$$

The gradient is useful because we can use it to build a linear approximation of $f(x)$ using the 1st order Taylor expansion

$$f(x^0 + d) = f(x^0) + \nabla f(x^0)^\top d + \epsilon(d)\|d\|_2, \tag{20}$$

where $\epsilon(d)$ is a continuous real valued such that

$$\lim_{d \to 0} \epsilon(d) = 0. \tag{21}$$

Furthermore, since $\epsilon(d)$ is continuous, given any constant $c > 0$ there exists $\delta > 0$ such that

$$\|d\| < \delta \quad \Rightarrow \quad \epsilon(d) < c. \tag{22}$$

We will use the expansion (20) to motivate and understand the steepest descent method later.

If $f(x)$ is twice differentiable, we refer to $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$ as the Hessian matrix:

$$\nabla^2 f(x) \quad \overset{\text{def}}{=} \quad \begin{bmatrix} \frac{\partial^2 f(z)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(z)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(z)}{\partial x_1 \partial x_3} & \cdots & \frac{\partial^2 f(z)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(z)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(z)}{\partial x_2 \partial x_2} & \frac{\partial^2 f(z)}{\partial x_2 \partial x_3} & \cdots & \frac{\partial^2 f(z)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(z)}{\partial x_n \partial x_1} & \frac{\partial^2 f(z)}{\partial x_n \partial x_2} & \frac{\partial^2 f(z)}{\partial x_n \partial x_3} & \cdots & \frac{\partial^2 f(z)}{\partial x_n \partial x_n} \end{bmatrix}.$$

If $f \in C^2$ (twice continuously differentiable) then

$$\frac{\partial^2 f(z)}{\partial x_i \partial x_j} = \frac{\partial^2 f(z)}{\partial x_j \partial x_i}, \quad \forall i, j \in \{1, \ldots, n\},$$

consequently the Hessian matrix is symmetric, that is, $\nabla^2 f(x) = \nabla^2 f(x)^\top$. Furthermore, for $f \in C^2$, with the Hessian matrix we can build a quadratic approximation to our function using the 2nd order Taylor expansion.

$$f(x^0 + d) = f(x^0) + \nabla f(x^0)^\top d + \frac{1}{2} d^\top \nabla^2 f(x^0) d + \epsilon(d)\|d\|_2^2. \tag{23}$$

We will use this expansion to motivate the Newton method later on.

To calculate the gradient and the Hessian, we need to use the chain-rule and product-rule.

Chain-rule. For every differentiable map $F(y) : \mathbb{R}^d \rightarrow \mathbb{R}^n$ and function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ we have that the gradient of the *composition*

$$f(F(x)) \;=\; f(F_1(x), \ldots, F_n(x)), \tag{24}$$

is given by

$$\nabla(f(F(x))) \;=\; \sum_{i=1}^{d} \frac{\partial}{\partial x_i}(f(F(x)))e_i \;=\; \sum_{i=1}^{d}\sum_{j=1}^{n} \frac{\partial}{\partial y_j} f(F(x)) \frac{\partial}{\partial x_i} F_j(x) e_i \;=\; \nabla F(x)^\top \nabla f(F(x)). \tag{25}$$

Example 2.1. Given a vector $a \in \mathbb{R}^n$ let $f(y) = y^\top a$. Show that

$$\nabla(f(F(x))) \;=\; \nabla F(x)^\top \frac{dy^\top a}{dy} \;=\; \nabla F(x)^\top a. \tag{26}$$

Chain-rule maps. Given any differentiable maps $F : \mathbb{R}^d \mapsto \mathbb{R}^m$ and $G : \mathbb{R}^m \mapsto \mathbb{R}^n$, the Jacobian of the composition

$$G(F(x)) \;=\; \left(G_1(F(x)), G_2(F(x)) \ldots, G_n(F(x))\right), \tag{27}$$

is given by

$$\nabla\left(G(F(x))\right) \;=\; \nabla G(F(x)) \nabla F(x). \tag{28}$$

Proof: One way to recall the chain-rule for multivariate functions is to reconstruct the rule using the definition of the gradient. That is

$$\begin{aligned}
\nabla(f(F(x))) \;&=\; \sum_{i=1}^{d} \frac{\partial}{\partial x_i}(f(F(x)))e_i \\
&=\; \sum_{i=1}^{d}\sum_{j=1}^{n} \frac{\partial f(F(x))}{\partial y_j} \frac{\partial F_j(x)}{\partial x_i} e_i \\
&=\; \sum_{j=1}^{n} \frac{\partial f(F(x))}{\partial y_j} \sum_{i=1}^{d} \frac{\partial F_j(x)}{\partial x_i} e_i \\
&=\; \sum_{j=1}^{n} \frac{\partial f(F(x))}{\partial y_j} \nabla F_j(x) \;=\; \nabla F(x)^\top \nabla f(F(x)).
\end{aligned} \tag{29}$$

Product-rule. For any two vector valued functions $F^1 : \mathbb{R}^d \mapsto \mathbb{R}^n$ and $F^2 : \mathbb{R}^d \mapsto \mathbb{R}^n$ we have that

$$\nabla(F^1(x)^\top F^2(x)) = \nabla F^1(x)^\top F^2(x) + \nabla F^2(x)^\top F^1(x). \tag{30}$$

Proof: The product-rule can be deduced from the chain-rule. For this, let $f(z,y) = z^\top y$, for $z, y \in \mathbb{R}^n$. Thus $F^1(x)^\top F^2(x) = f(F^1(x), F^2(x))$. By the chain-rule (25) we have that

$$\begin{aligned}
\nabla f(F^1(x), F^2(x)) &= \sum_{i=1}^{d} \left(\sum_{j=1}^{n} \frac{\partial}{\partial z_j} f(F^1(x), F^2(x)) \frac{\partial}{\partial x_i} F^1_j(x) + \sum_{j=n+1}^{2n} \frac{\partial}{\partial y_j} f(F^1(x), F^2(x)) \frac{\partial}{\partial x_i} F^2_j(x) \right) e_i \\
&= \sum_{i=1}^{d} \left(\sum_{j=1}^{n} F^2_j(x) \frac{\partial}{\partial x_i} F^1_j(x) + \sum_{j=1}^{n} F^2_j \frac{\partial}{\partial x_i} F^2_j(x) \right) e_i \\
&= \sum_{j=1}^{n} F^2_j(x) \nabla F^1_j(x) + \sum_{j=1}^{n} F^2_j \nabla F^2_j(x) \\
&= \nabla F^1(x)^\top F^2(x) + \nabla F^2(x)^\top F^1(x) \qquad (31)
\end{aligned}$$

Exercise 2.2. Let $f(x) = \frac{1}{2} x^\top A x + x^\top b + c$, where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Calculate the gradient and Hessian of $f(x)$.

Proof: Taking the gradient we have

$$\begin{aligned}
\nabla f(x) &= \tfrac{1}{2} \nabla (x^\top A x) + \nabla (x^\top b) + \nabla (c) \\
&\overset{(26)}{=} \tfrac{1}{2} \nabla (x^\top A x) + b \\
&\overset{(30)}{=} \tfrac{1}{2} \nabla (x)^\top A x + \tfrac{1}{2} \nabla (Ax)^\top x + b \\
&= \tfrac{1}{2} A x + \tfrac{1}{2} A^\top x + b \;=\; Ax + b,
\end{aligned}$$

where we used in the last two sets that $\nabla(Ax) = A$ by the definition of the Jacobian and that $A = A^\top$.

2.2 Level sets and geometry

For a fixed constant $C \in \mathbb{R}$ we refer to the following surface

$$\{x \in \mathbb{R}^n \,|\, f(x) = C\}, \qquad (32)$$

as the C level set of $f(x)$. Each level sets implicitly define a surface where $f(x)$ has the same value.

Exercise 2.3. Draw the level sets

$$\begin{aligned}
4 &= f(x_1, x_2), \\
9 &= f(x_1, x_2),
\end{aligned}$$

where $f(x_1, x_2,) = x_1^2 + x_2^2$.

The gradient direction is intimately linked to the geometry of the levels sets. Indeed, the gradient points orthogonal to the level sets. Thus we can also use the gradient to define the the tangent space, as follows.

Lemma 2.4. The tangent directions to a point on the level set of $f(x)$ are orthogonal to the gradient at that point. That is, for every

$$v \in T_f(x^0) \quad = \quad \left\{ v \in \mathbb{R}^n \, : \, v = \lim_{\substack{y \to x^0, \\ f(y) = f(x^0)}} \frac{y - x^0}{\|y - x^0\|_2} \right\}, \tag{33}$$

we have that $v^\top \nabla f(x^0) = 0$.

Proof: Let $y \in \mathbb{R}^n$ be such that $f(x^0) = f(y)$, consequently using (20) we have that

$$f(y) = f(x^0) + \nabla f(x^0)^\top (y - x^0) + \epsilon(y - x^0)\|y - x^0\|_2 = f(x^0).$$

Thus $\nabla f(x^0)^\top (y - x^0) + \epsilon(y - x^0)\|y - x^0\|_2 = 0$. Dividing by $\|y - x^0\|_2$ and taking the limit $y \to x^0$, gives that

$$0 = \lim_{y \to x^0} \left(\nabla f(x^0)^\top \frac{y - x^0}{\|y - x^0\|_2} + \epsilon(y - x^0) \right) \overset{(33)+(21)}{=} \nabla f(x^0)^\top v. \quad \square$$

Another way of phrasing the above result is that $y = f(x^0) + \nabla f(x^0)^\top d$ is the tangent line of $f(x)$ at $(x^0, f(x^0))$.

The gradient is also the direction that maximizes the local change in the objective function, while the negative gradient is the direction that minimizes the local change. This is the reason why gradient methods are often referred to as steepest descent methods.

Lemma 2.5 (Steepest Descent)**.** For a give $d \in \mathbb{R}^d$ we refer to

$$\Delta(d) = \lim_{s \to 0^+} \frac{f(x^0 + sd) - f(x^0)}{s}, \tag{34}$$

as the *local change* of $f(x)$ around x^0. Let $v = \nabla f(x^0) / \|\nabla f(x^0)\|_2$ be the normalized gradient. It follows that

$$\begin{aligned} v = \arg\max_{d \in \mathbb{R}^d} &\ \Delta(d) \\ \text{subject to } &\ \|d\|_2 = 1. \end{aligned} \tag{35}$$

That is, the normalized gradient is the direction maximizes the *local change* of $f(x)$ around x^0. Furthermore the negative normalized gradient minimizes the local change.

Proof: Using (20) we have that

$$f(x^0+sd)-f(x^0)=s\nabla f(x^0)^\top d+\epsilon(sd)s.$$

Dividing by s and taking the limit $s\to 0$ we have

$$\Delta(d)=\lim_{s\to 0^+}\frac{f(x^0+sd)-f(x^0)}{s}=\nabla f(x^0)^\top d+\lim_{s\to 0^+}\epsilon(sd)=\nabla f(x^0)^\top d.$$

Now using that $\|d\|_2=1$ together with the Cauchy inequality

$$-\|\nabla f(x^0)\|_2 \quad\leq\quad \Delta(d)=\nabla f(x^0)^\top d \quad\leq\quad \|\nabla f(x^0)\|_2. \tag{36}$$

The upper and lower bound is achieved when $d=\nabla f(x^0)/\|\nabla f(x^0)\|_2$ and $d=-\nabla f(x^0)/\|\nabla f(x^0)\|_2$, respectively. □

As a corollary, we have that the search direction is a *descent direction* if it has an obtuse angle with the gradient

Corollary 2.6 (Descent Condition)**.** If $d^\top\nabla f(x_0)<0$ then there exists $s>0$ such that

$$f(x_0+sd)<f(x_0).$$

Proof: From (36) we have that $\Delta(d)=\nabla f(x^0)^\top d<0$. Let $c=-\nabla f(x^0)^\top d>0$. Let $s>0$ be such that $\epsilon(sd)<\frac{c}{2}$. Consequently from (20) we have that

$$\frac{f(x^0+sd)-f(x^0)}{s}=\nabla f(x^0)^\top d+\epsilon(sd)\leq -\frac{c}{2}<0. \quad \square$$

2.3 Optimality conditions

Let $f\in C^2$ be twice continuously differentiable throughout this section.

In most cases it is impossible to solve a general optimization problem such as (17). A much more modest, and often attainable, objective is to find a local minima instead.

Definition 2.7. The point $x^*\in\mathbb{R}^n$ is a *local minima* of $f(x)$ is there exists $r>0$ such that

$$f(x^*)\leq f(x), \quad \forall\|x-x^*\|_2<r. \tag{37}$$

Thus a local minima point x^* is one where all neighbouring points are worst. This is often good enough for most applications (though not *all* applications). Finding local minima is made possible through the following necessary and sufficient conditions.

Theorem 2.8 (Necessary optimality conditions)**.** If x^* is a local minima of $f(x)$ then

1. $\nabla f(x^*)=0$

2. $d^\top \nabla^2 f(x^*) d \geq 0, \quad \forall d \in \mathbb{R}^n.$

Proof: The fact that $\nabla f(x^*) = 0$ follows from Corollary 2.6.

Now suppose there exists $d \in \mathbb{R}^n$ such that $d^\top \nabla^2 f(x^*) d < 0$. Suppose w.l.o.g that $\|d\|_2 = 1$. Using the second order Taylor expansion we have that

$$f(x^* + sd) \quad = \quad f(x^*) + \frac{s^2}{2} d^\top \nabla^2 f(x^*) d + \epsilon(sd) s^2.$$

Let $\bar{s} > 0$ be such that for $s \leq \bar{s}$ we have that $\epsilon(sd) < |d^\top \nabla^2 f(x^*) d| \big/ 4$. Dividing the above by s^2, for $s \leq \bar{s}$ we have that

$$\begin{aligned} \frac{f(x^* + sd)}{s^2} &= \frac{f(x^*)}{s^2} + \frac{1}{2} d^\top \nabla^2 f(x^*) d + \epsilon(sd) \\ &< \frac{f(x^*)}{s^2} + \frac{1}{4} d^\top \nabla^2 f(x^*) d, \end{aligned}$$

thus $f(x^* + sd) < f(x^*)$ for all $s \leq \bar{s}$ which contradicts the definition of local minima. □

Not only are the conditions in Theorem 2.8 necessary, with a slight modification, they are also sufficient.

Theorem 2.9 (Sufficient Local Optimality conditions)**.** If $x^* \in \mathbb{R}^n$ is such that

1. $\nabla f(x^*) = 0$

2. $d^\top \nabla^2 f(x^*) d > 0, \quad \forall d \in \mathbb{R}^n$ with $d \neq 0$,

then x^* is a local optima.

Proof: Let $d \in \mathbb{R}^n$ such that $\|d\|_2 = 1$. Because $\nabla^2 f(x^*)$ is positive definite, the smallest non-zero eigenvalue must also be strictly positive. Consequently

$$\lambda_{\min}(\nabla^2 f(x^*)) \leq d^\top \nabla^2 f(x^*) d, \quad \forall \|d\|_2 = 1.$$

Using the second-order Taylor expansion, we have that

$$\begin{aligned} f(x^* + d) &= f(x^*) + \frac{1}{2} d^\top \nabla^2 f(x^*) d + \epsilon(d) \|d\|_2^2 \\ &\geq \frac{\|d\|_2^2}{2} \lambda_{\min}(\nabla^2 f(x^*)) + \epsilon(d) \|d\|_2^2. \end{aligned}$$

Let $r > 0$ be such that every d with $\|d\| \leq r$ we have that

$$|\epsilon(d)| < \lambda_{\min}(\nabla^2 f(x^*))/4.$$

Thus for $\|d\| \leq r$ we have

$$\begin{aligned} f(x^* + d) &\geq f(x^*) + \frac{\|d\|_2^2}{2} \lambda_{\min}(\nabla^2 f(x^*)) + \epsilon(d) \|d\|_2^2 \\ &\geq f(x^*) + \frac{\|d\|_2^2}{4} \lambda_{\min}(\nabla^2 f(x^*)) > f(x^*). \quad \square \end{aligned}$$

Exercise 2.10. Let $f(x) = \frac{1}{2}x^\top Ax - x^\top b + c$, with A symmetric positive definite. How many local/global minimas can $f(x)$ have? Find a formula for the minima using only the *data* A and b.

Proof: By the sufficient conditions x^* is a local minima if $Ax = b$ and $A \succ 0$. Since $Ax = b$, there exists only one local minima.

2.4 Convex functions

In some cases, we can even find a global minima of (17). Indeed, for the *convex functions*, all local minima are global minima. We say a function is convex if

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y), \quad \forall x, y \in \mathbb{R}^d,\, t \in [0,\, 1]. \tag{38}$$

Theorem 2.11. If f is a convex function, then every local minima of f is also a global minima.

Proof: Let x^* be a local minima and suppose there exists $\bar{x} \in \mathbb{R}^n$ such that $f(\bar{x}) < f(x^*)$. Now consider the line segment $z = t\bar{x} + (1-t)x^*$ for $t \in [0,\, 1]$. By the definition of convexity we have that

$$f(z) = f((1-t)\bar{x} + tx^*) \leq (1-t)f(\bar{x}) + tf(x^*) < (1-t)f(x^*) + tf(x^*) = f(x^*). \tag{39}$$

This shows that x^* cannot be a local minima. Indeed, for any $r >$ with $\|\bar{x} - x^*\|_2 \leq r > 0$, we have that by choosing $t =\leq 1 - r/\|\bar{x} - x^*\|_2$ we have that

$$\|z - x^*\|_2 = (1-t)\|\bar{x} - x^*\|_2 \leq r.$$

Yet from (39) we have that $f(z) < f(x^*)$. A contraction. Thus there exists no $\bar{x}$ with $f(\bar{x}) < f(x^*)$. □

When f is twice differentiable, there are three equivalent ways of defining convexity.

Theorem 2.12. If f is twice continuously differentiable, then the following three statements are equivalent

$$\begin{aligned} f(tx + (1-t)y) &\leq tf(x) + (1-t)f(y), \quad \forall x, y \in \mathbb{R}^d,\, t \in [0,\, 1]. && (40)\\ f(y) &\geq f(x) + \nabla f(x)^\top (y - x), \quad \forall x, y \in \mathbb{R}^d. && (41)\\ 0 &\leq d^\top \nabla^2 f(x) d, \quad \forall x, d \in \mathbb{R}^d. && (42) \end{aligned}$$

Proof: We only prove that (40)⇒ (41)⇒ (42). The remaining proof of (42)⇒ (41)⇒ (40) is left as an exercise.

((40)⇒ (41)) We can deduce (41) from (40) by dividing by t and re-arranging

$$\frac{f(y + t(x-y)) - f(y)}{t} \leq f(x) - f(y).$$

Now taking the limit $t \to 0$ gives

$$\langle \nabla f(y), x - y \rangle \leq f(x) - f(y).$$

((41)⇒ (42)) First we prove this holds for 1–dimensional functions $f : \mathbb{R} \to \mathbb{R}$. From (41) we have that

$$\begin{array}{rcl} f(y) & \geq & f(x) + f'(x)(y - x), \\ f(x) & \geq & f(y) + f'(y)(x - y). \end{array}$$

Combining the above two we have that

$$f'(x)(y - x) \leq f(y) - f(x) \leq f'(y)(y - x).$$

Dividing by $(y - x)^2$ we have

$$\frac{f'(y) - f'(x)}{y - x} \geq 0, \quad \forall x, y, x \neq y.$$

It remains to take the limit. To then extend to any n–dimensional function, use that

$$\left.\frac{d^2 f(x + tv)}{dv^2}\right|_{t=0} = v^\top \nabla^2 f(x) v \geq 0, \forall v \neq 0.$$

□

We also say that $f(x)$ is strictly convex when $d^\top \nabla^2 f(x) d > 0, \quad \forall d \in \mathbb{R}^d$ with $d \neq 0$.

Exercise 2.13 (Homework)**.** Prove Theorem (2.12) for quadratic functions $f(x) = \frac{1}{2} x^\top A x + x^\top b + c$. What can we say about A when f is convex or strictly convex?

2.5 Gradient method

Since the negative gradient is the direction that minimizes the local change, it is only natural to use the negative gradient as a descent direction. This suggests the following method

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

But the question still remains how to choose s^k the stepsize? One natural strategy is to choose s^k that results in the maximum descent. That is,

$$s^k = \arg\min_{s \geq 0} f(x^k + s d^k). \tag{43}$$

Finally, we also need a criteria to determine when we should stop iterating. Typically the user will choose a *tolerance to error* $\epsilon > 0$, and then stop the algorithm either when we are at an approximate stationary point

$$\|\nabla f(x^k)\|_2 \leq \epsilon \tag{44}$$

Algorithm 2 Steepest Descent

```
Choose $x^0 \in \mathbb{R}^n$.
while $\|\nabla f(x^k)\|_2 > 0\epsilon$ or $f(x^{k+1}) - f(x^k) \leq \epsilon$ do
    Calculate $d^k = -\nabla f(x^k)$
    Calculate $s^k = \arg\min_{s \geq 0} f(x^k + sd^k)$
    Update $x^{k+1} = x^k + s^k d^k$.
```

or when the rate in decrease is small

$$f(x^{k+1}) - f(x^k) \leq \epsilon. \tag{45}$$

This gives the steepest descent method in Algorithm (2).

The rate of convergence in (47) may be slow. Indeed, it depends on how far the smallest and largest singular are from each other. Perhaps the issue is that we used a fixed stepsize instead of the "best" stepsize (43)? It turns out that also this greedy strategy of choosing the best step size in (43) can also lead slow convergence. This is partly due to a phenomenon called *zig-zagging*. When using the optimal stepsize (43), sequential gradients are orthogonal to one another, that is

$$\nabla f(x^{k+1})^\top \nabla f(x^{k+1}) = 0. \tag{46}$$

To prove this, note that by the definition of s^k in (43) we have that

$$\frac{d}{ds} f(x^k - s\nabla f(x^k))\Big|_{s=s^k} = 0.$$

Using the chain-rule we have that

$$\frac{d}{ds} f(x^k - s\nabla f(x^k))\Big|_{s=s^k} = -s^k \nabla f(x^k - s\nabla f(x^k))^\top \nabla f(x^k) = 0.$$

Thus so long as $s^k \neq 0$, then (46) holds. This zigzagging can make the global convergence of steepest descent painfully slow. One way to fix this zigzagging is to use a constant stepsize. Yet another that works well in practice is to substitute the exact line search with an approximate line search such as the *Backtracking line search*, see Algorithm 3. This line search method tries increasingly smaller stepsizes until a sufficient decrease in the function is obtained.

Algorithm 3 Backtracking Line Search

```
Choose $\bar{\alpha} > 0, \rho, c \in (0,\, 1)$. Set $\alpha = \bar{\alpha}$
while $f(x^k + \alpha d^k) \leq f(x^k) + c\alpha \nabla f(x^k)^\top d^k$ do
    Update $\alpha = \rho\alpha$.
```

To give some theoretical insight, we now prove that Algorithm 2 converges for a constant stepsize and quadratic functions

Theorem 2.14. Let $f(x) = \frac{1}{2}x^\top Ax - x^\top b + c$. If we choose a fixed stepsize of $s^k = 1/\,\sigma_{\max}(A)$ then the iterates of Algorithm 2 converge according to

$$\|\nabla f(x^{k+1})\|_2 \leq \left(1 - \frac{\sigma_{\min}(A)}{\sigma_{\max}(A)}\right)^k \|\nabla f(x^0)\|. \tag{47}$$

Proof:

$$\begin{aligned}
\nabla f(x^{k+1}) &= Ax^{k+1} - b \\
&= A(x^k - s\nabla f(x^k)) - b \\
&= A(x^k - s(Ax^k - b)) - b \\
&= Ax^k - b - sA(Ax^k - b) \\
&= (I - sA)\nabla f(x^k).
\end{aligned}$$

Taking norms

$$\|\nabla f(x^{k+1})\|_2 \leq \|I - sA\|_2 \|\nabla f(x^k)\|.$$

By choosing $s = 1/\sigma_{\max}(A)$ we have that $I - sA$ is symmetric positive definite and

$$\|I - sA\|_2 = 1 - s\sigma_{\min}(A) = 1 - \frac{\sigma_{\min}(A)}{\sigma_{\max}(A)} < 1.$$

Robert: Homework: Prove this last step!

Thus finally

$$\|\nabla f(x^{k+1})\|_2 \leq \left(1 - \frac{\sigma_{\min}(A)}{\sigma_{\max}(A)}\right) \|\nabla f(x^k)\| \leq \left(1 - \frac{\sigma_{\min}(A)}{\sigma_{\max}(A)}\right)^k \|\nabla f(x^0)\|.$$

Algorithm 2 also converges for all *smooth* functions, that is when there exists $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in \mathbb{R}^n. \tag{48}$$

See the notes here `http://users.ece.utexas.edu/~cmcaram/EE381V_2012F/Lecture_4_Scribe_Notes.final.pdf` for a similar proof for all smooth functions.

2.6 Newton's method

Using the second order Taylor expansion we can motivate the Newton method. At each iteration, the *Newton Method* minimizes the local quadratic approximation

$$q_k(x) = f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top \nabla^2 f(x^k)(x - x^k), \tag{49}$$

as a proxy for minimizing $f(x)$. Assuming $f(x)$ is strictly convex, the stationary point x^{k+1} characterized by $\nabla_x q_k(x) = 0$ of (49) is given by

$$\nabla_x q_k(x) = \nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k) = 0.$$

Isolating x^{k+1} we have

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

The advantage that Newton's method has over the steepest descent is that it converges quadratically when x is close to the minimum.

Theorem 2.15. Let $f(x)$ be a μ–strongly convex function, that is, we have a global lower bound on the Hessian given by

$$\nabla^2 f(x) \succeq \mu I, \quad \forall x \in \mathbb{R}^n. \tag{50}$$

Furthermore, if the Hessian is also Lipschitz

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2 \tag{51}$$

then Newton's method converges locally and quadratically fast according to

$$\|x^{k+1} - x^*\|_2 \leq \frac{L}{2\mu}\|x^k - x^*\|_2^2. \tag{52}$$

In particular if $\|x^0 - x^*\|_2 \leq \frac{\mu}{L}$, then for $k \geq 1$ we have that

$$\|x^{k+1} - x^*\|_2 \leq \frac{1}{2^{2^k}}\frac{\mu}{L}. \tag{53}$$

Proof:

$$\begin{aligned}
x^{k+1} - x^* &= x^k - x^* - \nabla^2 f(x^k)^{-1}\left(\nabla f(x^k) - \nabla f(x^*)\right) \\
&= x^k - x^* - \nabla^2 f(x^k)^{-1}\int_{s=0}^1 \nabla^2 f(x^k + s(x^* - x^k))(x^k - x^*)ds \quad \text{(Mean value theorem)} \\
&= \nabla^2 f(x^k)^{-1}\int_{s=0}^1 \left(\nabla^2 f(x^k) - \nabla^2 f(x^k + s(x^* - x^k))\right)(x^k - x^*)ds
\end{aligned}$$

Taking norms we have that

$$\begin{aligned}
\|x^{k+1} - x^*\|_2 &\leq \|\nabla^2 f(x^k)^{-1}\|_2 \int_{s=0}^1 \|\nabla^2 f(x^k) - \nabla^2 f(x^k + s(x^* - x^k))\|_2 \|x^k - x^*\|_2 ds \\
&\overset{(51)+(50)}{\leq} \frac{L}{\mu}\int_{s=0}^1 s\|x^k - x^*\|_2^2 ds \\
&= \frac{L}{2\mu}\|x^k - x^*\|_2^2.
\end{aligned}$$

Finally if $\|x^0 - x^*\|_2 \leq \frac{\mu}{L}$, then by induction and assuming that (53) holds we have that

$$\|x^{k+1} - x^*\| \quad \leq \quad \frac{L}{2\mu}\|x^k - x^*\|_2^2 \quad \leq \quad \frac{L}{2\mu}\frac{1}{2^{2^k}}\frac{1}{2^{2^k}}\left(\frac{\mu}{L}\right)^2 \quad < \quad \frac{1}{2^{2^{k+1}}}\frac{\mu}{L},$$

which concludes the induction proof. □

The assumptions on for this proof can be relaxed, since we only require the Hessian is Lipschitz and lower bounded in a $\frac{\mu}{2L}$–ball around x^*.

3 Nonlinear programming with constraints

For this section, I highly recommend reading Chapter 12 in [3].

The objectives of this section are

1. To introduce nonlinear optimization with contraints and provide a geometric intuition
2. Establish necessary first order optimality conditions which can be easily verified with linear algebra.

Let f, g_i and h_j be C^1 continuous functions, for $i = 1, \ldots, m$ and $j = 1, \ldots, p$. Consider the *constrained* optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad \text{for } i \in I. \\ & h_j(x) = 0, \quad \text{for } j \in J, \end{aligned} \tag{54}$$

where $I = \{1, \ldots, m\}$, $J = \{1, \ldots, p\}$, $g_i(x) \leq 0$ are the *inequality constraints* and $h_j(x) = 0$ are the *equality constraint.* If a point $x \in \mathbb{R}^n$ satisfies all the inequality and equality constraints we say that x is a *feasible* point. We refer to the set of all feasible points as the *feasible set* which we denote by

$$X \overset{\text{def}}{=} \{x \in \mathbb{R}^n \, : \, g_i(x) \leq 0, \, h_j(x) = 0, \quad \text{for } i = 1, \ldots, m, \text{ and } j = 1, \ldots, p.\}.$$

If x is such that $g_i(x) = 0$, we say that the ith inequality constraint is *saturated* or *active* at x.

Adding constraints can make the problem easier to solve as compared to the unconstrained problem (17), since now we do not need to search the whole of $\mathbb{R}^n$ for a solution, but rather only a subset X. Indeed, if $X = \{x_0\}$ then clearly the solution is x_0. But constraints can add many difficulties. Indeed, even with smooth differentiable functions g_i and h_j, the frontier of X may be non-smooth (linear constraints define a polyhedron!). Often the difficulty of dealing with constraints is guaranteeing feasibility and descent directions that are feasible. That is why we focus next on characterizing feasible descent directions. See the examples in the beginning of Chapter 12 in [3] for examples on the difficulties that constraints introduce.

Theorem 3.1 (Existence)**.** If the feasible set X is bounded and non-empty, then there exists a solution to (54).

Proof: Given that the sets $\mathbb{R}_- = [-\infty, 0]$ and $\{0\}$ are closed, by the continuity of g_i and h_j we have that X is closed. Indeed,

$$X = \left(\bigcap_{i=1}^{m} g_i^{-1}([-\infty, 0]) \right) \cap \left(\bigcap_{j=1}^{p} h_j^{-1}(\{0\}) \right),$$

and thus is a finite intersection of closed sets. By assumption X is bounded, thus it is compact. By the continuity of f we have that $f(X)$ is also compact (The Extreme value theorem). Consequently there exists a minimum in $f(X)$.

Definition 3.2. We say that $f : \mathbb{R}^n \to \mathbb{R}$ is coercive if $\lim_{\|x\|\to\infty} f(x) = \infty$.

Theorem 3.3. If X is non-empty and f is coercive, then there exists a solution to (54)

Proof: Let $x' \in X$. Let $B_r(x) = \{x : \|x\| \leq r\}$. Since f is coercive, there exists r such that for each x with $\|x\| \geq r$ we have that $f(x) \geq f(x')$. Thus clearly the minimum of f is in $B_r(x)$. Since $B_r(x)$ is bounded and closed, we have that $B_r(x) \cap X$ is bounded and closed. Again by the extreme value theorem, $f(x)$ attains its minimum in $B_r(x) \cap X$, which is also the minimum in X. □

3.1 Admissable and Feasible directions

To design iterative methods for solving constained optimization problems, we need to know how to move from one given point and still remain within the feasible set X. For instance if X was a polyhedra, then we would say that d is a *feasible* or an *admissible* direction at $x_0 \in X$ if there exists $\epsilon > 0$ such that $x_0 + td \in X$ for all $0 \leq t \leq \epsilon$. To account for the case that the frontier of the feasible set are not just straight lines, we need to consider a more general notation of feasible directions.

Definition 3.4. We say that d is an *admissible* direction at $x_0 \in X$ if there exists a C^1 differentiable curve $\phi : \mathbb{R}_+ \to \mathbb{R}^n$ such that

1. $\phi(0) = x_0$
2. $\phi'(0) = d$
3. There exists $\epsilon > 0$ such that $t \leq \epsilon$ we have $\phi(t) \in X$

We denote by $A(x_0)$ the set of admissable directions at x_0.

To prove that a certain $d \in \mathbb{R}^n$ is an admissable direciton we will almost always use the 1st order Taylor expansion of $\phi(t)$ given by

$$\phi(t) = x_0 + td + t\vec{\epsilon}(t), \tag{55}$$

where $\vec{\epsilon}(t) \in \mathbb{R}^n$ such that $\lim_{t\to 0} ||\vec{\epsilon}(t)||_2 = 0$. We will often compose differentiable functions $f : \mathbb{R}^n \to \mathbb{R}$ with this curve. In this setting, we will often make use of the following representation of the first order Taylor expansion

Lemma 3.5. Let $\phi : \mathbb{R}_+ \to \mathbb{R}^n$ be a C^1 curve as defined in Definition 3.4. Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable. Then the first order Taylor expansion of the composition $f(\phi(t))$ around x_0 can be written as

$$f(\phi(t)) = f(x_0) + td^\top \nabla f(x_0) + t\hat{\epsilon}(t), \tag{56}$$

where $\lim_{t\to 0} \hat{\epsilon}(t) = 0$.

Proof:

Robert: Easy proof:

Since both f and ϕ are C^1, their composition is also C^1. Thus $f(\phi(t))$ has a first order Taylor expansion around $t = 0$ gives

$$f(\phi(t)) = f(\phi(0)) + t\frac{df(\phi(t))}{dt}|_{t=0} + t\epsilon(t).$$

Now it is just a matter of plugging in $\phi(0) = x_0$ and

$$\frac{df(\phi(t))}{dt}|_{t=0} \;=\; (\phi'(t)^\top \nabla f(\phi(t)))|_{t=0} \;=\; (d^\top \nabla f(x_0)).$$

Robert: Unecessarily difficult proof:

Using the first order expansion of ϕ followed by the first order expansion of f we have that

$$\begin{aligned} f(\phi(t)) &\overset{(55)}{=} f(x_0 + td + t\vec{\epsilon}(t)) \\ &= f(x_0) + t(d + \vec{\epsilon}(t))^\top \nabla f(x_0) + \|t(d+\vec{\epsilon}(t))\|\epsilon(\|t(d+\vec{\epsilon}(t))\|) \\ &= f(x_0) + td^\top \nabla f(x_0) + t\underbrace{\left(\|d+\vec{\epsilon}(t)\|\epsilon(t\|d+\vec{\epsilon}(t)\|) + \vec{\epsilon}(t)^\top \nabla f(x_0)\right)}_{\overset{\text{def}}{=}\hat{\epsilon}(t)}. \end{aligned}$$

Now it is easy to see that $\lim_{t\to 0}\hat{\epsilon}(t) = 0$ since $\epsilon(t\|d+\vec{\epsilon}(t)\|) \to 0$ and $\vec{\epsilon}(t)^\top \nabla f(x_0) \to 0$.

Thus $\phi(t)$ can be replaced by a line segment up to terms that go to zero faster than t.

Let $I_0(x_0) = \{i : g_i(x_0) = 0,\, i \in I\}$ be the indexes of saturated inequalities.

A necessary condition on a direction being admissable is given in the following.

Proposition 3.6. If $d \in A(x_0)$ is a admissable direction then

1. For every $i \in I(x_0)$ we have that $d^\top \nabla g_i(x^0) \le 0$.

2. For every $j \in J$ we have that $d^\top \nabla h_j(x^0) = 0$.

Let $B(x_0)$ be the set of directions that satisfy the above two conditions. Thus $A(x_0) \subset B(x_0)$.

Proof: 1. Let $i \in I$ and let $d_1 = d/\|d\|$. If $g_i(x_0) = 0$. Let $\phi(t)$ be the curve associated to d_1. Consider the 1st order Taylor expansion of g_i around x_0 in the d_1 direction which is

$$\begin{aligned} g_i(\phi(t)) &\overset{(56)}{=} g_i(x_0) + td^\top \nabla g_i(x_0) + t\epsilon(t) \\ &= td^\top \nabla g_i(x_0) + t\epsilon(t) \\ &\leq 0, \end{aligned}$$

where we used $g_i(\phi(t)) \leq 0$ for t sufficiently small. Dividing by t we have that

$$d^\top \nabla g_i(x^0) + \epsilon(t) \leq 0.$$

Letting $t \to 0$ we have that $d^\top \nabla g_i(x^0) \leq 0$.
2. Using the first order Taylor expansion of h_j around x_0 we have that

$$h_j(\phi(t)) \overset{(56)}{=} h_j(x_0) + td^\top \nabla h_j(x_0) + t\epsilon(t) = td^\top \nabla h_j(x_0) + t\epsilon(t) = 0. \tag{57}$$

Dividing by t and then taking the limit as $t \to 0$ gives $d^\top \nabla h_j(x^0) = 0$. □

Note that $B(x_0)$ is a cone, and we will refer to $B(x_0)$ as the cone of feasible directions. Because the cone $B(x_0)$ is easy to work with, we would like to use $B(x_0)$ instead $A(x_0)$. Indeed, it only depends on linear constraints involving gradients. But sometimes $B(x_0)$ is not equivalent to $A(x_0)$ when there exists certain degeneracies. For instance, if

$$h_1(x) = (x_1 - x_2 - 2)^2 = 0,$$

or

$$h_1(x) = (x_1^2 + x_2^2 - 2)^2 = 0,$$

then $\nabla h_1(x) = 0$ for all feasible points, and thus we cannot use $\nabla h_1(x)$ to describe feasible directions.

To exclude the possibility of such degeneracies, we impose the *Constraint qualifications*, which are the conditions that ensure we can use $B(x_0)$ instead of $A(x_0)$.

Definition 3.7. We say that constraint qualifications hold at x_0 if for every $d \in B(x_0)$ there exists a sequence $(d_t)_{t=1}^{\infty} \in A(x_0)$ such that $d_t \to d$.

An example where the constraint qualifications do not hold, and consequently $B(x_0) \neq A(x_0)$ is given now.

Example 3.8.

Robert: Broken example!

Consider the constraints given by

$$\begin{cases} x^2 + (y-1)^2 \geq 1 \\ x \leq 1 \\ y \geq 0. \end{cases}$$

That is, $g_1(x,y) = 1-x^2-(y-1)^2$, $g_2(x,y) = x-1$ and $g_3(x,y) = -y$. At the point $(x^0, y^0) = (0,0)$ it is easy to show that $B(x^0, y^0) = \{(t,\, 0)\; :\; \forall t \in \mathbb{R}\}$ while $A(x^0, y^0) = \{(t,\, 0)\; :\; \forall t \geq 0\}$. Thus the constraint qualification cannot hold, and $B(x^0, y^0)$ is a bad substitute for $A(x^0, y^0)$.

The following theorem justifies why constraint qualifications are important.

Theorem 3.9 (Necessary conditions)**.** Let x^* be a local minimum of (54) and suppose the constraint qualification holds at x^*. It follows that for every $d \in B(x^*)$ we have that $\nabla f(x^*)^\top d \geq 0$. In other words, all direction in the feasible cone are not descent directions.

Proof: Let $d_k \in A(x_*)$ be a sequence such that $d_k \to d$. Let ϕ_k be the curve associated to d_k. According to Lemma 3.5 we have that the first order Taylor expansion of $f(\phi_k(t))$ can be written as

$$f(\phi_k(t)) = f(x_*) + t\nabla f(x_*)^\top d_k + t\epsilon_k(t). \tag{58}$$

Since x_* is a local minima, there exists T for which $t \leq T$ we have that $f(x*) \leq f(\phi_t(\alpha_t))$. Consequently

$$t\nabla f(x_*)^\top d_k + t\epsilon_k(t) \geq 0, \quad \text{for } t \leq T.$$

Dividing by t and taking the limit we have

$$0 \leq \lim_t \nabla f(x_*)^\top d_k + \epsilon_k(t) = \nabla f(x_*)^\top d_k.$$

Taking the limit in k concludes the proof. □

3.2 Characterizing the constraint qualification

Robert: This section is not necssary! I didn't teach it last year

First we show that sufficient conditions to characterize an admissable direction. We then move of to use this to establish sufficient and easily verifiable conditions which when verified imply the constraint qualification holds.

Lemma 3.10. Suppose that h_j is an affine function for $j \in J$. Let $x^0 \in X$ and $d \in \mathbb{R}^n$ be such that

$$\begin{aligned} d^\top \nabla g_i(x^0) &< 0, \quad \text{for } i \in I(x^0) \\ d^\top \nabla h_j(x^0) &= 0, \quad \text{for } j \in J. \end{aligned}$$

Then d is a admissable direction, that is $d \in A(x_0)$.

Proof: Let $\phi(t) = x^0 + td$. Since h_j is affine we have that

$$h_j(\phi(t)) = h_j(x^0) + td^\top \nabla h_j(x^0).$$

Furthermore, given that x^0 is feasible we have that $h_j(x^0) = 0$ and by assumption $d^\top \nabla h_j(x^0) = 0$, consequently $h_j(\phi(t)) = 0$. Furthermore, for $i \in I(x_0)$ we have that

$$g_i(\phi(t)) = g_i(x_0) + td^\top \nabla g_i(x^0) + t\epsilon(t).$$

Since $g_i(x^0) = 0$ and $d^\top \nabla g_i(x^0) < 0$, there exists $t > 0$ sufficiently small so that $g_i(\phi(t)) \le 0$. Thus $x^0 + td \in X$ for $t > 0$ sufficiently small, which proves that d is a feasible direction. □

We now give sufficient condition for the constraint qualifications to hold.

Lemma 3.11. Suppose h_j is affine for all $j \in J$. Let $x_0 \in X$. If there exists $\tilde{d}$ such that

$$\begin{aligned} \tilde{d}^\top \nabla g_i(x^0) &< 0, \quad \text{for } i \in I(x^0) \\ \tilde{d}^\top \nabla h_j(x^0) &= 0, \quad \text{for } j \in J, \end{aligned}$$

then the constraint qualification holds at x^0.

Proof: Let $d \in B(x^0)$. We will now show that

$$d_t = td + (1-t)\tilde{d},$$

for $t \in [0, 1[$ we have that $d_t \in A(x_0)$ and for $t \to 0$ we have that d_t converges to d, which in turn shows that the constraint qualifications hold. Indeed, for $i \in I(x_0)$ we have that

$$d_t^\top \nabla g_i(x^0) = t \underbrace{d^\top \nabla g_i(x^0)}_{\le 0,\ d \in B(x^0)} + (1-t)\tilde{d}^\top \nabla g_i(x^0) < 0.$$

Furthermore

$$d_t^\top \nabla h_j(x^0) = td^\top \nabla h_j(x^0) + (1-t)\tilde{d}^\top \nabla h_j(x^0) = 0.$$

Consequently, by Lemma 3.10, we have that for all $t \in [0, 1[$ we have shown that $d_t \in A(x_0)$. Now taking any sequence $t_n \to 1$ we have that $d_t \to d$, thus by definition the constraint qualifications hold at x_0. □

We will now use the previously Lemma to develop sufficient condition for the constraint qualifications to hold in the entire feasible set X.

Proposition 3.12. If g_i for $i \in I$ are convex and h_j for $j \in J$ are affine and there exists a single point $\tilde{x} \in X$ such that

$$g_i(\tilde{x}) < 0, \quad h_j(\tilde{x}) = 0 \quad \forall i \in I,\ j \in J, \tag{59}$$

then the constraint qualification holds for all feasible points in X.

Proof: Let $x^0 \in X$ and let $\tilde{x}$ satisfy (59). Using the 1st order Taylor expansion around $\tilde{x}$ we have that

$$0 \overset{(59)}{>} g_i(\tilde{x}) \overset{\text{convexity}}{\geq} g_i(x^0) + \nabla g_i(x^0)^\top(\tilde{x} - x^0).$$

If $i \in I(x^0)$ then $g_i(x^0)$ and from the above we have that $\nabla g_i(x^0)^\top(\tilde{x} - x^0) < 0$. Let $\tilde{d} \overset{\text{def}}{=} (\tilde{x} - x^0)$. Furthermore, for the equality constraints we have that

$$h_j(\tilde{x}) \overset{\text{affine.}}{=} h_j(x^0) + \nabla h_j(x^0)^\top \tilde{d}.$$

Since $h_j(x^0) = 0$ and thus $\nabla h_j(x^0)^\top \tilde{d} = 0$. This shows that $\tilde{d}$ satisfies the conditions of Lemma 3.11, and thus the constraint qualifications hold at x^0. □

Yet another alternative characterization of constraint qualifications for a given x_0, which is easier to verify, is given in the following proposition.

Proposition 3.13. Suppose that h_j are affine functions for $j \in J$. For a given feasible point $x_0 \in X$ we have that the set

$$\{\nabla g_i(x_0) \,:\, i \in I_0(x_0)\} \cup \{\nabla h_j(x_0) \,:\, j \in J\},$$

is a linearly independent set, then the constraint qualifications hold at x_0.

Proof: Let $x^0 \in X$ and consider the following two Linear Programs

$$\begin{aligned} \max_{\lambda,\mu} z &= \sum_{i \in I(x^0)} \lambda_i + \sum_{j \in J} 0 \times \mu_j. \\ \text{subject to } &\sum_{j \in J} \mu_j \nabla h_j(x^0) - \sum_{i \in I(x^0)} \lambda_i \nabla g_i(x^0) = 0, \\ &\lambda_i \geq 0, \text{for } i \in I(x^0). \end{aligned} \tag{P}$$

and

$$\begin{aligned} &\min_d w \equiv 0 \\ \text{subject to } &d^\top \nabla g_i(x^0) \leq -1, , \text{for } i \in I(x^0), \\ &d^\top \nabla h_j(x^0) = 0, \text{for } j \in J. \end{aligned} \tag{D}$$

It is not hard to show that (P) is the primal and (D) is its associated dual problem. Indeed, to see this, first we right (P) in the standard form by introducing variables $\mu^+, \mu^- \in \mathbb{R}^p$ with $\mu = \mu^+ - \mu^-$ and including the constraints $\mu^+, \mu^- \geq 0$. Then write $c = (\underbrace{1, \ldots, 1}_{|I(x^0)|}, \underbrace{0, \ldots, 0}_{2p})$, $b = 0 \in \mathbb{R}^n$, and

$$A = \left[-\nabla g_1(x^0), \ldots, -\nabla g_m(x^0), \nabla h_1(x^0), \ldots, \nabla h_p(x^0), -\nabla h_1(x^0), \ldots, -\nabla h_p(x^0).\right]$$

From (D) the dual constraints are given by

$$A^\top d \geq c \Rightarrow \begin{cases} d^\top \nabla h_j(x^0) \geq 0, \\ -d^\top \nabla h_j(x^0) \geq 0, \\ -d^\top \nabla g_i(x^0) \geq 1, \quad \text{for } i \in I(x^0),\ j \in J. \end{cases}$$

Which exactly the constraint of (D). The rest follows by examining (D).

Since (P) is non-empty, since it admits the zero solution. Now suppose that (P) admits a solution λ_i^*, μ^* that is not zero. In which case, we have that the gradients of the constraints must be linearly independent. This contractdicts our assumption, thus $(\lambda_i^*, \mu^*) = (0,0)$ is the optimal solution. Thus (P) is bounded and by strong duality Theorem 1.5 we have that (D) is feasible. Let $\tilde{d}$ be a feasible solution to (D). Note that $\tilde{d}$ satisfies the assumptions of Lemma 3.11, consequently the constraint qualification holds at x_0. □

3.3 Lagrange's condition

Consider the problem (54) but without inequality constraints, that is,

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_j(x) = 0, \quad \text{for } j \in J, \end{aligned} \tag{60}$$

The following theorem gives us necessary conditions for a point to be a local minima that requires only using linear algebra.

Theorem 3.14. Let $x^* \in X$ be a local minima and suppose that the constraint qualifications hold at x^* for (54). It follows that the gradient of the objective is a linear combination of the gradients of constraints at x^*, that is, there exists $\mu_j \in \mathbb{R}$ for $j \in J$ such that

$$\nabla f(x^*) = \sum_{j \in J} \mu_j \nabla h_j(x^*). \tag{61}$$

Proof: Let $E = \text{span}\left(\{\nabla h_1(x^*), \ldots, \nabla h_p(x^*)\}\right)$. Let us re-write $\nabla f(x^*) = y + z$ where $y \in E$ and $w \in E^\perp$. By definition

$$-z^\top \nabla h_j(x^*) = 0, \quad \forall j \in J.$$

Thus $-z \in B(x^*)$. Consequently by Theorem 3.9 we have that

$$-z^\top \nabla f(x^*) \geq 0.$$

Thus

$$-z^\top \nabla f(x^*) = -z^\top y - \|z\|_2^2 = -\|z\|_2^2 \geq 0.$$

Consequently $z = 0$ and $\nabla f(x^*) = y \in E$. □

3.4 Karush, Kuhn and Tuckers condition

The extension of Lagrange's condition to allow to our general problem (54) with inequality constraints is known at the Karush, Kuhn Tuckers

Theorem 3.15 (Necessary conditions)**.** Let $x^* \in X$ be a local minima and suppose that the constraint qualifications hold at x^* for (60). It follows that there exists $\mu_j \in \mathbb{R}$ and $\lambda_i \in \mathbb{R}_+$ for $j \in J$ and $i \in I(x^*)$ such that

$$\nabla f(x^*) = \sum_{j \in J} \mu_j \nabla h_j(x^*) - \sum_{i \in I(x^*)} \lambda_i \nabla g_i(x^*). \tag{62}$$

Proof: From Theorem 3.9 we know that for every $d \in B(x^*)$ we have that $\nabla f(x^*)^\top d \geq 0$. That is, said more explicitly, we have that for every $d \in \mathbb{R}^n$ that satisfies

$$\begin{aligned} -d^\top \nabla g_i(x^*) &\geq 0, \quad \text{for } i \in I(x^*) \\ d^\top \nabla h_j(x^*) &= 0, \quad \text{for } j \in J, \end{aligned}$$

we have that $d^\top \nabla f(x^*) \geq 0$. This statement fits perfectly the 2nd version of Farkas Theorem 3.18. That is, let

$$A = [-\nabla g_1(x^*), \ldots - \nabla g_m(x^*)] \quad \text{and} \quad B = [\nabla h_1(x^*), \ldots \nabla h_p(x^*)],$$

and $b = \nabla f(x^*)$. Thus for every

$$d \in \{d \,:\, A^\top d \geq 0, \quad B^\top d = 0\}$$

we have that $d^\top b \geq 0$. By Farkas Theorem 3.18 we have that this statement can be equivalently re-written as there exists $(\lambda,\, \mu) \in P$ where

$$P = \{(\lambda, \mu) \,:\, A\lambda + B\mu = b, \quad \lambda \geq 0\}.$$

Through which we conclude that the set

$$P = \{\mu \in \mathbb{R}^p,\, \lambda \in \mathbb{R}^m_+ \,:\, \sum_{j \in J} \mu_j \nabla h_j(x^*) - \sum_{i \in I(x^*)} \lambda_i \nabla g_i(x^*) = \nabla f(x^*)\},$$

is non-empty, which concludes the proof. □

Theorem 3.16 (Sufficient conditions)**.** Let f and g_i for $i \in I$ be convex functions. Let h_j be linear for $j \in J$. Suppose the constraint qualifications hold at $x^* \in X$ and the KKT conditions are verified. Then x^* is a local minima

Proof: Since the KKT conditions hold, there exist $\mu_j \in \mathbb{R}$ and $\lambda_i \in \mathbb{R}_+$ for $j \in J$ and $i \in I(x^*)$ such that (62) holds. Let $x \in X$. Since $f(x)$ is convex, we have that

$$\begin{aligned} f(x) \quad &\geq \quad f(x^*) + \nabla f(x^*)^\top (x - x^*) \\ &\overset{(62)}{=} \quad f(x^*) + \sum_{j \in J} \mu_j \nabla h_j(x^*)^\top (x - x^*) - \sum_{i \in I(x^*)} \lambda_i \nabla g_i(x^*)^\top (x - x^*). \end{aligned} \tag{63}$$

Using the linearity of each h_j we have that

$$\nabla h_j(x^*)^\top (x - x^*) = h_j(x) - h_j(x^*) = 0.$$

Since each g_i is convex, we have that

$$\nabla g_i(x^*)^\top (x - x^*) \leq g_i(x) - g_i(x^*).$$

Plugging the above into (63) gives $f(x) \geq f(x^*)$.

I will now illustrate the above theorem, on the board, with only inequality constraints.

First, what is the largest sphere that fits in an ellipsoid?

Exercise 3.17.

$$\begin{aligned} &\min -x^2 - y^2 \\ &\text{subject to } ax^2 + by^2 \leq 1, \end{aligned}$$

where $a > b > 0$.

Using the KKT conditions and assuming that the constraint is not active, that is $ax^2 + by^2 < 1$, we quickly arrive at the solution $(x, y) = (0, 0)$.

Now assuming the constraint is active, that is $ax^2 + by^2 = 1$, we have the KKT conditions

$$\begin{aligned} 2x &= 2a\lambda x \\ 2y &= 2b\lambda y. \end{aligned}$$

From which we conclude that either 1) $\lambda = a^{-1}$, $y = 0$ and $x = \pm a^{-1/}$ or 2) $\lambda = b^{-1}$, $x = 0$ and $y = \pm b^{-1/2}$. In the first case, we have that $f(x, y) = -x^2 - y^2 = -a^{-1}$. Alternatively in the second case we have $f(x, y) = -b^{-1}$. Since $-b^{-1} < -a^{-1} \leq 0$, we have that $(x, y) = (0, \pm b^{-1/2})$ are the two minimum. What is the maximum?

3.5 The constrained descent method

In the practical exercise we will consider only inequality constraints, that is

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad \text{for } i \in I. \end{aligned} \tag{64}$$

Algorithm 4 Descent Algorithm

```
Choose $x^0 \in X$. Set $k = 0$.
while KKT($x^k$) conditions not verified do
    Find $d$ such that $d^\top \nabla f(x^k) \leq 0$ and $d^\top \nabla g_j(x^k) \leq 0$.        # Find feasible direciton
    Find $s \in \mathbb{R}_+$ such that $f(x^k + sd) < f(x^k)$ and $x^k + sd \in X$        # Stay in set
    $x^{k+1} = x^k + sd$        # Take a step
    $k = k + 1$
```

We will design an iterative method that fits the following format in Algorithm 4.

In finding a direction that satifies line 3, we can solve a minimization problem such as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & d^\top \nabla f(x^k) \\ \text{subject to} \quad & d^\top \nabla g_j(x^k) \leq 0, \quad \text{for } i \in I(x^k) \\ & d^\top d = 1 \end{aligned} \tag{65}$$

References

[1] Marco Chiarandini. “Linear and Integer Programming Lecture Notes”. In: (2015).

[2] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* 1st ed. Springer Publishing Company, Incorporated, 2014.

[3] J Nocedal and S J Wright. *Numerical Optimization.* Ed. by Peter Glynn and Stephen M Robinson. Vol. 43. Springer Series in Operations Research 2. Springer, 1999. Chap. 5, pp. 164–75.

Appendix

Theorem 3.18 (2nd Version of Farkas)**.** Consider the set

$$P = \{(\lambda, \mu) \,:\, A\lambda + B\mu = b, \quad \lambda \geq 0\}$$

and

$$Q = \{y \,:\, A^\top y \geq 0, \quad B^\top y = 0\}.$$

The set P is non-empty if and only every $y \in Q$ is such that $b^\top y \geq 0$.

Proof: (1) If P non-empty then $y \in Q \Rightarrow b^\top y \geq 0$: Let $\mu = z^+ - z^-$ where $z^+, z^- \geq 0$. Then clearly P is non-empty if and only if

$$\hat{P} = \{(\lambda, z^+, z^-) \,:\, A\lambda + Bz^+ - Bz^- = b, \quad \lambda, z^+, z^- \geq 0\},$$

is non-empty. The dual program of $\hat{P}$ if we consider a zero objective function $(z = 0)$ is given by

$$\min_{y} \; y^\top b$$
$$\text{subject to } A^\top y \geq 0, \quad B^\top y \geq 0, \quad -B^\top y \geq 0. \tag{66}$$

see Lemma 3.20 for how to deduce this version of duality. The feasible region of (66) is equivalent to Q. By the weak duality Lemma 1.4 we have that if there exists a feasible primal point then for all feasible y, that is, all $y \in Q$ we have $b^\top y \geq c^\top(\lambda, z^+, z^-) = 0$.

(2)If $y \in Q \Rightarrow b^\top y \geq 0$ **then** P **non-empty:** Clearly $0 \in Q$ and thus the dual (66) is feasible. Since $y \in Q$ implies that $b^\top y \geq 0$, we have that (66) is feasible and bounded. This proves by duality that P is feasible. □

Proof: [Using 1st Farkas Theorem] Let $\mu = z^+ - z^-$ where $z^+, z^- \geq 0$. Then clearly P is non-empty if and only if

$$\hat{P} = \{(\lambda, z^+, z^-) \; : \; A\lambda + Bz^+ - Bz^- = b, \quad \lambda, z^+, z^- \geq 0\},$$

is non-empty. Applying the 1st Farkas Theorem 3.19, we have that $\hat{P}$ is non-empty if every $y \in \hat{Q}$ implies that $y^\top b \geq 0$ where

$$\hat{Q} = \{y \; : \; A^\top y \geq 0, \quad B^\top y \geq 0, \quad -B^\top y \geq 0\}.$$

Clearly $Q = \hat{Q}$ which concludes the proof. □

Theorem 3.19 (Farkas Version 1)**.** Consider the set

$$P = \{x \; : \; Ax = b, \quad x \geq 0\}$$

and

$$Q = \{y \; : \; A^\top y \geq 0\}.$$

The set P is non-empty if and only every $y \in Q$ is such that $b^\top y \geq 0$.

Proof: Any text on optimization.

Lemma 3.20 (Duality 2nd version)**.** Consider the following LP with equality constraints

$$\max_{x} \; c^\top x$$
$$\text{subject to } Ax = b,$$
$$x \geq 0, \tag{LP}$$

Then the dual is given by

$$\begin{aligned} &\min_y \; y^\top b \\ &\text{subject to } A^\top y \geq c. \end{aligned} \tag{DP}$$

Proof: First we re-write (LP) as

$$\begin{aligned} &\max_x \; c^\top x \\ \text{subject to } &Ax \leq b, \\ &-Ax \leq -b, \\ &x \geq 0, \end{aligned} \tag{LP}$$

The dual program (D) of the above is given by

$$\begin{aligned} &\min \; w^\top b - z^\top b \\ \text{subject to } &A^\top w - A^\top z \geq c, \\ &w, z \geq 0. \end{aligned}$$

Substituting $y = w - z$ gives the result.