

1 12th of December 2018 — A. Frangioni

1.1 Quadratic problem with linear equality constraints

Let A be a matrix in $M(m, n, \mathbb{R})$, where $m < n$ (otherwise the system is either fully determined or impossible) and $rk(A) = m$. The constrained quadratic problem may be written as

$$\min \left\{ \frac{1}{2} x^T Q x + q x \quad : \quad A x = b \right\}$$

where $Q \succeq 0$.

A way to solve this problem is through Karush Kuhn Tucker system:

$$\begin{array}{l} \text{(a)} \quad \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix} \\ \text{(b)} \end{array} \quad (1.1)$$

where the first row (a) says that the gradient is a linear combination of the normals of the gradient of the constraints and the second one is just feasibility.

Everything is linear here. This system is symmetric, although indefinite, because it contains many 0s.

We are left with solving the KKT system:

REDUCED KKT: in this method we first add the hypothesis of non-singularity to the matrix Q so we get the following

$$\begin{cases} Qx + A^T \mu = -q & \text{(a)} \\ Ax = b & \text{(b)} \end{cases}$$

Multiply (a) by AQ^{-1}

$$\begin{cases} AQ^{-1}Qx + AQ^{-1}A^T \mu = -AQ^{-1}q & \text{(a)} \\ Ax = b & \text{(b)} \end{cases}$$

Multiply (b) by -1 and add to it (a)

$$\begin{cases} Ax + AQ^{-1}A^T \mu = -AQ^{-1}q & \text{(a)} \\ \cancel{Ax} + AQ^{-1}A^T \mu - \cancel{Ax} = -AQ^{-1}q - b & \text{(b)} \end{cases}$$

Multiply (a) by A^{-1}

$$\begin{cases} x + Q^{-1}A^T \mu = -Q^{-1}q & \text{(a)} \\ AQ^{-1}A^T \mu = -AQ^{-1}q - b & \text{(b)} \end{cases}$$

Isolate x

$$\begin{cases} x = -Q^{-1}(A^T\mu + q) & (a) \\ AQ^{-1}A^T\mu = -AQ^{-1}q - b & (b) \end{cases}$$

Notice that $0 \preceq AQ^{-1}A^T = M \in M(m, \mathbb{R})$ and may be much smaller than the original one, since its size depends on the number of constraints. The issue here is that the matrix M is less sparse than both A and Q .

NULL SPACE METHOD: In this method we need no assumption on Q .

First of all we rearrange the matrix A in order to have a small square matrix A_B and then the rest of the columns (A_N): $A = [A_B, A_N]$, $x = [x_B, x_N]$, $\det(A_B) \neq 0$.

Replacing A in (b) we get $A_Bx_B + A_Nx_N = b \iff x_B = A_B^{-1} \cdot (b - A_Nx_N)$, hence resulting in $x = Dx_N + d$, where

$$d = \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} -A_B^{-1}A_N \\ I \end{bmatrix} \in M(m, n - m)$$

Notice that D is a basis of the **null space** (or kernel) of A and it is not mandatory to build it like we did above, it is only important to obtain a basis of the kernel.

Let us multiply (a) by D^T and obtain

$$\begin{aligned} D^T(Qx + A^T\mu) &= -D^Tq \\ D^TQx + D^TA^T\mu &= -D^Tq \\ D^TQx + (AD)^T\mu &= -D^Tq \\ D^TQ(Dx_N + d) &= -D^Tq \end{aligned} \tag{1.2}$$

Where in the last step we applied the definition $x = Dx_N + d$ and hence, $(D^TQD)x_N = -D^T(Qd + q)$.

We term $H = D^TQD \in M(n - m, \mathbb{R})$ the reduced Hessian of the problem and notice that whenever the number of constraints is close to the number of variables this matrix is very small.

It is important to note that in order to solve an equality constrained problem we can choose either the reduced KKT method or the null space method, depending on the structure of our problem.

1.2 Inequality constrained problems

The problem, in this case, is written as follows

$$(P) \min\{f(x) : Ax \leq b\}$$

1.2.1 Projected gradient method

The intuition behind this kind of algorithm is that we are interested in finding as direction for the line search not the opposite of the gradient (because then the step size should be put to 0 if we lie on the boundaries, see Figure 1.1), but it is enough to pick a direction which has a negative scalar product with the gradient.

The rationale is to pick the direction that minimizes the norm of the difference with the gradient. Formally

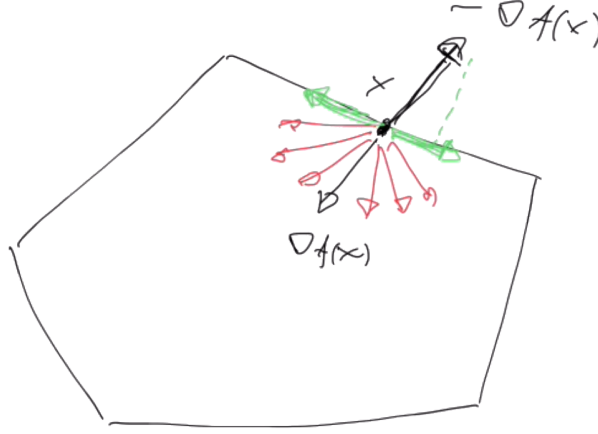


FIGURE 1.1: Geometric idea of how to choose the direction in case of lying on the boundary.

$$d = \operatorname{argmin}\{\|\nabla f(x) - d\|^2 : d \in \mathcal{D}_X(x)\}$$

where $\mathcal{D}_X(x)$ is the feasible cone. Notice that if the difference between the gradient and the direction is 0 it means that we can stop, since the descent direction would bring us outside the feasible set.

A more formal definition of the feasible cone $\mathcal{D}_X(x) = \{d \in \mathbb{R}^n : A_{\mathcal{A}(x)}d \leq 0\}$, where $\mathcal{A}(x)$ is the set of all the active constraints.

For sake of clarity we denote $\bar{A} = A_{\mathcal{A}(x)}$.

At this point we are ready to project the problem in such a way that inequality constraints become equality constraints:

$$\min \left\{ \frac{1}{2} \|\nabla f(x) + d\|^2 = \frac{1}{2} d^T I d + \nabla f(x) d : \bar{A} d = 0 \right\}$$

Note that in this case the Hessian is the identity matrix $I \succ 0$, hence the KKT conditions become simpler:

$$\begin{cases} Qx + A^T \mu = -q & \iff d + A^T \mu = -\nabla f(x) \\ Ad = 0 \end{cases}$$

since $x = d$, $A = I$, $b = \nabla f(x)d$, so we get

$$\begin{aligned} d &= -\nabla f(x) - A^T \mu \\ A d &= -A \nabla f(x) - A A^T \mu \\ A A^T \mu &= -A \nabla f(x) \end{aligned} \tag{1.3}$$

resulting in

$$\begin{cases} Qx + A^T \mu = -q & \Longleftrightarrow & d + A^T \mu = -\nabla f(x) \\ A d = 0 \end{cases}$$

Therefore, we need to solve a system in μ and then we have the direction. If the number of active constraints is small, solving the system is very fast.

Fact 1.1. *We can restrict to solve*

$$\begin{cases} \mu = -(\bar{A} \bar{A}^T)^{-1} \bar{A} \nabla f(x) \\ d = (I - \bar{A}^T (\bar{A} \bar{A}^T)^{-1} \bar{A}) (-\nabla f(x)) \end{cases}$$

Proof. \bar{A} has full row rank, then is non singular and may be inverted.

$$\begin{cases} \mu = -(\bar{A} \bar{A}^T)^{-1} \bar{A} \nabla f(x) \\ d = -\nabla f(x) + \bar{A}^T (\bar{A} \bar{A}^T)^{-1} \bar{A} \nabla f(x) \end{cases} \Longleftrightarrow \begin{cases} \mu = -(\bar{A} \bar{A}^T)^{-1} \bar{A} \nabla f(x) \\ d = (I - \bar{A}^T (\bar{A} \bar{A}^T)^{-1} \bar{A}) (-\nabla f(x)) \end{cases}$$

□

If the set of active constraints contains some linear dependent ones, we take the maximal subset of our constraints such that the matrix \bar{A} has full rank.

This procedure is formalized in Algorithm 1.1, where at each stage we remove one of the constraints that lead to a negative μ , keeping a set of linearly independent constraints.

In the 14-th line of the algorithm we find that our step size is takes as the minimum step size among the ones that satisfy some subsets of the constraints.

ALGORITHM 1.1 Pseudocode for projected gradient method for quadratic functions.

```

1: procedure PGM( $f, A, b, x, \varepsilon$ )
2:   for (; ;) do
3:      $B \leftarrow$  maximal  $\subseteq \mathcal{A}(x)$  s.t.  $\text{rank}(A_B) = |B|$ ;
4:     for (; ;) do
5:        $d \leftarrow (I - A_B^T(A_B A_B^T)^{-1} A_B)(-\nabla f(x))$ ;
6:       if  $\langle \nabla f(x), d \rangle \leq \varepsilon$  then
7:          $\mu_B \leftarrow -(A_B A_B^T)^{-1} A_B \nabla f(x)$ ;
8:          $\mu_i \leftarrow 0 \ \forall i \notin B$ ;
9:         if  $\mu_B \geq 0$  then return
10:        end if
11:         $h \leftarrow \min\{i \in B : \mu_i < 0\}$ ;
12:        end if
13:         $\bar{\alpha} \leftarrow \min\{\alpha_i = (b_i - A_i x)/A_i d : A_i d > 0, i \notin B\}$ ;
14:         $x \leftarrow x + \alpha d$ ;
15:        if  $\bar{\alpha} > 0$  then
16:          break;
17:        end if
18:         $k \leftarrow \min\{i \notin B : A_i d > 0 : \alpha_i = 0\}$ ;
19:         $B \leftarrow B \cup \{k\}$ ;
20:      end for
21:       $\alpha \leftarrow \text{Line\_Search}(f, x, d, \bar{\alpha})$ ;
22:       $x \leftarrow x + \alpha d$ ;
23:    end for
24: end procedure

```

Fact 1.2. *The following holds:*

1. $d = 0 \wedge \mu \geq 0 \implies x$ optimal (from KKT);
2. $d = 0 \wedge \exists h \in B$ s.t. $\mu_h < 0 \implies \exists x' \in \{x \in \mathbb{R}^n : A_{B \setminus \{h\}} x = b_{B \setminus \{h\}}, A_h x \leq b_h\}$ s.t. $f(x') < f(x)$;
3. $d \neq 0$ descent direction: $H = I - A_B^T[A_B A_B^T]^{-1} A_B$ symmetric and idempotent $HH = H^T H = H \implies \langle d, \nabla f(x) \rangle < 0$.

Proof.

trivial;

- The intuition is that if we remove h from B next time we will have a descent direction.
 $\exists d$ s.t. $A_{B \setminus \{h\}} d = 0 \wedge A_h d < 0 \implies \langle \nabla f(x), d \rangle = \langle -\mu A_B, d \rangle = -\mu_h A_h d < 0$;

- $\langle d, \nabla f(x) \rangle = \langle -H\nabla f(x), \nabla f(x) \rangle = -\nabla f(x)^T H \nabla f(x) = -(H\nabla f(x))^T H \nabla f(x).$ \square

Once we found the optimal face we move inside that face and we are dealing with a steepest descent, which is slow.

At a certain point of the execution the set B of the active constraints will stabilize and become the one of the optimal solution.

At this point the smart thing to do is to use the KKT conditions, since we know the set of active constraints. This idea is pursued in Section 1.2.3.

1.2.2 Projected gradient method with box constraints

Given the non linear minimum problem of the following shape

$$(P) \min \left\{ f(x) : l \leq x \leq u \right\}$$

a slightly different version of the projected gradient method forces the algorithm to put to 0 the component of the direction which would bring us to go outside the feasible region. This intuition is formalized in Algorithm 1.2.

ALGORITHM 1.2 Pseudocode for projected gradient method for quadratic functions in the case of box constraints.

```

1: procedure PGMBC( $f, l, u, x, \varepsilon$ )
2:    $d = -\nabla f(x)$ ;
3:    $\bar{\alpha} = \infty$ ;
4:   for ( $i = 1 \dots n$  s.t.  $d_i \neq 0$ ) do
5:     if ( $d_i < 0$ ) then
6:       if ( $x_i = l_i$ ) then
7:          $d_i = 0$ ;
8:       else
9:          $\bar{\alpha} \leftarrow \min\{\bar{\alpha}(x_i - l_i)/d_i\}$ ;
10:      end if
11:    else
12:      if ( $x_i = u_i$ ) then
13:         $d_i = 0$ ;
14:      else
15:         $\bar{\alpha} \leftarrow \min\{\bar{\alpha}(u_i - x_i)/d_i\}$ ;
16:      end if
17:    end if
18:    if ( $\langle \nabla f(x), d \rangle \leq \varepsilon$ ) then return
19:    end if
20:     $\alpha \leftarrow \text{Line\_Search}(f, x, d, \bar{\alpha})$ ;
21:     $x \leftarrow x + \alpha d$ ;
22:  end for
23: end procedure

```

Notice that we can assume that $l_i < u_i \forall i$, because otherwise that component would be fixed.

1.2.3 Active-set method for quadratic programs

Let us be given the following minimum problem

$$\min \left\{ \frac{1}{2} x^T Q x + q x : A x \leq b \right\}$$

where an important hypothesis is that the problem is quadratic, once we know $\mathcal{A}(x_*)$.

We start from a certain point, such that the constraints are satisfied as equalities. According to KKT conditions, a solution \bar{x} is optimal if \bar{x} is feasible and $\mu \geq 0$. Otherwise, if μ is not positive we eliminate something from the active set and start again. In the case of x unfeasible, we know the descent direction, we only need to revise the step size.

All this machinery is formalized in Algorithm 1.3.

ALGORITHM 1.3 Pseudocode for active set method for quadratic functions.

```

1: procedure ASMQP( $Q, q, A, b, x, \varepsilon$ )
2:   for ( $B \leftarrow \mathcal{A}(x); ;$ ) do
3:     solve ( $P_B$ )  $\min\{\frac{1}{2}x^T Qx + qx : A_B x = b_B\}$  for  $(\bar{x}, \bar{\mu}_B)$ ;
4:     if ( $A_i \bar{x} \leq b_i \forall i \notin B$ ) then
5:       if ( $\mu_B \geq 0$ ) then return
6:       end if
7:        $h \leftarrow \min\{i \in B : \mu_i < 0\}$ ;
8:        $B \leftarrow B \setminus \{h\}$ ;
9:       continue;
10:    end if
11:     $d \leftarrow \bar{x} - x$ ;
12:     $\bar{\alpha} \leftarrow \min\{\alpha_i = (b_i - A_i x)/A_i d : A_i d > 0, i \notin B\}$ ;
13:     $x \leftarrow x + \bar{\alpha} d$ ;
14:     $B \leftarrow \mathcal{A}(x)$ ;
15:  end for
16: end procedure

```

Notice that this algorithm allows easy solving of the same problem, under box constraints, because we end up having two sets: L , indexes of variables fixed to the lower bound, and U , indexes of variables fixed to the upper bound.

$\{1, \dots, n\} \setminus (L \cup U)$ is the set of the indexes of the free variables.

Thanks to this consideration the problem to solve becomes

$$\min \left\{ \frac{1}{2} x_F^T Q_{FF} x_F + (q_F + u_U^T Q_{UF}) x_F \right\} \left[+ \frac{1}{2} x_U^T Q_{UU} x_U + q_U u_U \right]$$

1.2.4 Frank-Wolfe method

Let us take a non-linear function and the following problem

Supposing f is convex we can make a first order model and minimize it over our constraints.

In this case we use the right constraints and an approximation of the function.

ALGORITHM 1.4 Pseudocode for Frank-Wolf method for non linear functions.

```

1: procedure FWM( $f, A, b, x, \varepsilon$ )
2:   while ( $\|\nabla f(x)\| > \varepsilon$ ) do
3:      $\bar{x} \leftarrow \operatorname{argmin} \{ \langle \nabla f(x), y \rangle : Ay \leq b \}$ ;
4:      $d \leftarrow \bar{x} - x$ ;
5:      $\alpha \leftarrow \operatorname{Line\_Search}(f, x, d, 1)$ ;
6:      $x \leftarrow x + \alpha d$ ;
7:   end while
8: end procedure

```

Provided that the computation at line 3 of Algorithm 1.4 is performed efficiently, then either $d = 0$ and we are in the optimum or $d > 0$ and we are moving toward the optimum.

The linear model is below the function in the quadratic case, hence we get a lower bound.