Quadratic Programming with equality constraints

Claudio Fantasia
Politecnico di Torino
Student id: s319911
s319911@studenti.polito.it

***Abstract*—This review presents the different approaches used in Quadratic Programming (QP) for solving an optimization problem in the restricted case of equality constraints only.**
Three different numerical methods with different properties and results will be issued: MINRES, Schur complement approach and Null Space Method.

I. Introduction

Quadratic programming (QP) is the process of solving certain optimization problems that involves minimizing or maximizing a quadratic objective function subject to a set of linear constraints.
The usual formulation of QP with equality constraints is:

$$\min_x \frac{1}{2}\mathbf{x}^\intercal Qx + \mathbf{c}^\intercal x$$
$$s.t \quad Ax = b$$

where:

- $Q \in R^{n\times x}$: symmetric positive semidefinite matrix
- $c, x \in R^n$
- $A \in R^{m\times n}$, $b \in R^m$

The computational cost of QP is highly dependent on the optimization problem that need to be solved.
Different objective function and different constraints will lead to various formulation of Q, c, A, b
Rewriting the QP formulation in the Lagragian form we obtain:

$$L(x, \lambda) = \min_x \frac{1}{2}\mathbf{x}^\intercal Qx + \mathbf{c}^\intercal x + \lambda^\intercal(Ax - b) \tag{1}$$

Thanks to this formulation, the minimum of the objective function can be found using **Karush-Kuhn-Tucker condition (KKT)** deriving for x and λ

$$\nabla_x = \; Qx^* + c + A^\intercal\lambda^* = 0 \tag{2}$$

$$\nabla_\lambda = \; Ax^* - b = 0 \tag{3}$$

We introduce a more compound formulation in order to have a single linear system

$$K = \begin{bmatrix} Q & A^\intercal \\ A & 0 \end{bmatrix}, \; w = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \; d = \begin{bmatrix} -c \\ b \end{bmatrix}$$

Obtaining:

$$Kw = d \tag{4}$$

Thanks to 2 and 3 we find the candidate solutions to QP. To assure that the solution is unique the **Lagrangian Multiplier Theorem** must be satisfied and in order to do that we have to assure that A is full rank and K is not singular

II. Optimization problem

$$\min_{x\in R^n} \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n-1} x_i x_{i+1} + \sum_{i=1}^{n} x_i \tag{5}$$

$$\begin{aligned} x_1 + x_{1+K} + x_{1+2K} + ... &= 1 \\ x_2 + x_{2+K} + x_{2+2K} + ... &= 1 \\ &\vdots \\ x_K + x_{2K} + x_{3K} + ... &= 1 \end{aligned}$$

The problem has been solved using as parameters n = $\left[10^4, 10^5\right]$ and K = $[100, 500]$ (i.e. m can either be 100 or 500
By our problem definition we can derive some insightful informations. First we notice that A is full rank and Q is symmetric positive definite. But mostly important matrix K is non-singular (thus invertible) and its inertia is (n, m, 0).
Notice that Q and so K are high dimensional sparse matrix with a lot of non-diagonal values equal to zero. A classical computer is not able to store the matrix in its RAM and to prevent **Out-of-Memory** error, the matrices have been initialized using MATLAB sparse functions.
In the interest of comparing the different proposed approaches the metric used are the norm of (2) and (3). An optimal solution would be value 0 to both the equation. More close we are to zero, more accurate our approach is.

III. Schur complement approach

The Schur complement approach is a technique that offers a different way to formulate (2) and (3) towards a problem formulation that is easier to compute.
Notice that Schur complement is a viable solution only if Q is positive definite. Fortunately this is our case.
We define the Schur complement as $\hat{Q} = AQ^{-1}A^\intercal$ obtaining the new formulation:

$$\hat{Q}\lambda^* = -(AQ^{-1}c + b) \tag{6}$$

$$x^* = Q^{-1}(-c - A^\intercal\lambda^*) \tag{7}$$

The Schur complement is useful because it lead us to a solution of a square problem in (6) and the dimension of

$\hat{Q} \in R^{m\times m}$ are way smaller that the initial dimension. Of course this formulation is enhanced by the fact that Q is easily invertible, otherwise if Q is not sparse like in our case, the computational cost of the inverse can became considerable.
Notice that for reducing the computational cost, matrix $\hat{Q}$ has been computed using the MATLAB backslash operator instead of the command $inv(Q)$.
The results are in the table below

	$n = 10^4$	$n = 10^5$
$K = 100$	$5.3167 \cdot 10^{-2}$	$15.711 \cdot 10^{-2}$
$K = 500$	$11.116 \cdot 10^{-2}$	$167.42 \cdot 10^{-2}$

TABLE I: Schur results: $computational\, time(seconds)$

	$n = 10^4$	$n = 10^5$
$K = 100$	$< eps$	$< eps$
$K = 500$	$< eps$	$< eps$

TABLE II: Schur results: KKT_gradX_norm

	$n = 10^4$	$n = 10^5$
$K = 100$	$9.5178 \cdot 10^{-5}$	$27090 \cdot 10^{-5}$
$K = 500$	$4.5717 \cdot 10^{-5}$	$12400 \cdot 10^{-5}$

TABLE III: Schur results: KKT_gradL_norm

	$n = 10^4$	$n = 10^5$
$K = 100$	99.9999	102.7092
$K = 500$	499.999	502.772

TABLE IV: Schur results: $objective\, function\, value$

IV. NULL SPACE METHOD

Also the Null space method offer a different problem formulation that is easier to compute. Unlike the Schur complement, the null space does not require Q to be SPD, but Q can also be Semi-Positive Definite.
Null space method is based on the assumption that we know a particular solution $\hat{x}$ to the linear system (3) and we also assume to have a full rank matrix $Z \in R^{n\times(n-m)}$ such that the columns of Z form a basis for $Ker(A)$ (i.e $AZ = 0$). This will be useful in a moment to claim that $Z^\intercal QZ$ is SPD.
Given that we introduce a vector $v \in R^{n-m}$ and we can rewrite the variable x as:

$$x = Zv + \hat{x} \tag{8}$$

Replacing (8) in (2) and after some computation we obtain the equation to find the optimal value v (thus the optimal x) and λ for the objective function.

$$Z^\intercal QZv^* = -Z^\intercal(c + Q\hat{x}) \tag{9}$$

$$AA^\intercal\lambda^* = -A(c + Qx^*) \tag{10}$$

As it has been claimed before $Z^\intercal QZ$ is SPD so it can be decomposed using **Cholesky factorization** and obtain an undemanding linear system computation.
Notice that $\hat{x}$ is an arbitrary vector in R^n and we can arbitrarily assume that some elements are 0 and the result formulation would be:

$$\hat{x} = \begin{bmatrix} A^{-1}b \\ 0 \end{bmatrix},$$

The results are presented in the table below

	$n = 10^4$	$n = 10^5$
$K = 100$	$2.3113 \cdot 10^{-1}$	$899.8923 \cdot 10^{-1}$
$K = 500$	$1.3152 \cdot 10^{-1}$	$53.6396 \cdot 10^{-1}$

TABLE V: Null space results: $computational\, time(seconds)$

	$n = 10^4$	$n = 10^5$
$K = 100$	$4.3332 \cdot 10^{-14}$	$19.4850 \cdot 10^{-14}$
$K = 500$	$1.5716 \cdot 10^{-14}$	$14.5920 \cdot 10^{-14}$

TABLE VI: Null space results: KKT_gradX_norm

	$n = 10^4$	$n = 10^5$
$K = 100$	$2.4248 \cdot 10^{-15}$	$8.2441 \cdot 10^{-15}$
$K = 500$	$2.9079 \cdot 10^{-15}$	$7.9859 \cdot 10^{-15}$

TABLE VII: Null space results: KKT_gradL_norm

	$n = 10^4$	$n = 10^5$
$K = 100$	100.000	100.000
$K = 500$	500.000	500.000

TABLE VIII: Null space results: $objective\, function\, value$

V. MINIMAL RESIDUAL METHOD

The **Minimal Residual Method** (MINRES) is an iterative method used to solve linear systems of equations where the matrix is symmetric.
This method is also particularly useful when dealing with large sparse symmetric matrix, as it offers advantages in terms of memory usage and computational efficiency
Our purpose is to solve (4) using MINRES since matrix decomposition of K is not possible due to the sparse nature of the matrix. LDL(or LU) decompositions would likely lead to the **fill-in phenomenon**
The criterion used for MINRES are $tol = 10^{-7}$ and $maxit = 2000$. Taking as example the most challenging parameters for MINRES (i.e $n = 10^{-5}, K = 500$) the computational time was 0.73 seconds and the algorithm converged in 157 iterations.
Notice that the convergence rate of MINRES method it depends by $2\left(\frac{\sqrt{cond(K)}-1}{\sqrt{cond(K)}+1}\right)^k$ with k number of iterations.
In our case the conditioning number of K is very high ($cond(K) = 3.73 \cdot 10^{10}$) and we have to be careful to not set tol too small.
The results are presented in the table below

	$n = 10^4$	$n = 10^5$
$K = 100$	496	52
$K = 500$	1530	157

TABLE IX: MINRES results: $number\ of\ iteration$

	$n = 10^4$	$n = 10^5$
$K = 100$	$7.917 \cdot 10^{-2}$	$16.417 \cdot 10^{-2}$
$K = 500$	$24.363 \cdot 10^{-2}$	$66.840 \cdot 10^{-2}$

TABLE X: MINRES results: $computational\, time(seconds)$

	$n = 10^4$	$n = 10^5$
$K = 100$	$1.003 \cdot 10^{-5}$	$3.098 \cdot 10^{-5}$
$K = 500$	$1.023 \cdot 10^{-5}$	$3.139 \cdot 10^{-5}$

TABLE XI: MINRES results: KKT_gradX_norm

	$n = 10^4$	$n = 10^5$
$K = 100$	$1.985 \cdot 10^{-7}$	$3.943 \cdot 10^{-7}$
$K = 500$	$1.810 \cdot 10^{-7}$	$5.252 \cdot 10^{-7}$

TABLE XII: MINRES results: KKT_gradL_norm

	$n = 10^4$	$n = 10^5$
$K = 100$	100.000	100.000
$K = 500$	500.000	500.000

TABLE XIII: MINRES results: $objective\ function\ value$

VI. Discussion

As explained in the previous sections, every method issued has its advantages and disadvantages. For example Schur complement require Q to be Positive definite, while Null space require Q to be at least Semi-Positive definite.
In general Schur complement is more likely to be picked, because it need to solve a smaller linear system that depends on the number of equality constraints, while the Null space depends on the creation of Z that can be bad conditioned, if A is not highly sparse like in our case issued.
Even though Schur complement is the fastest method, it looks like it is the least accurate, obtaining the highest value for KKT_gradL_norm and the $objective\ function\ value$ does not converge to its real value.
In general we can't obtain a value too accurate using MINRES because setting a tolerance too small will increase significantly the number of iteration and its elapsed time.
It is important to notice that the more accurate approach is the Null space as we can see in VI and VII, even though the Null Space is one of the method with slower convergence as we can see in Fig 1.
Notice that the computational time for computing the solution with $n = 10^5$ $K = 100$ using Null Space is way higher than the all other parameters setting and this may be further investigated substituting the optimization problem faced.

[1]

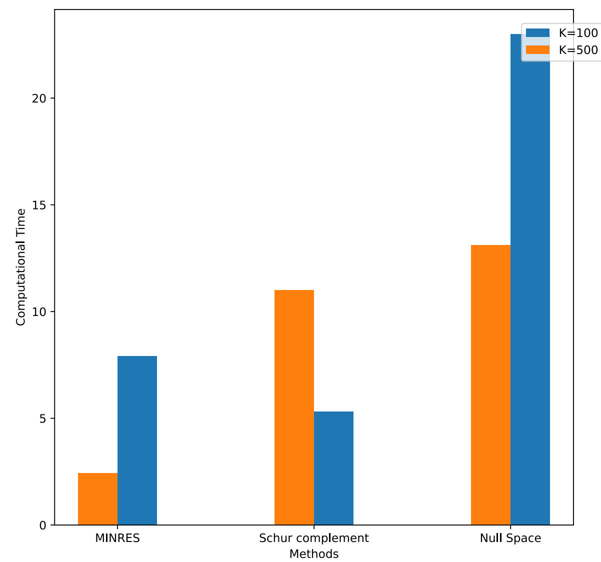

Fig. 1: Elapsed time among the three methods for $n = 10^4$

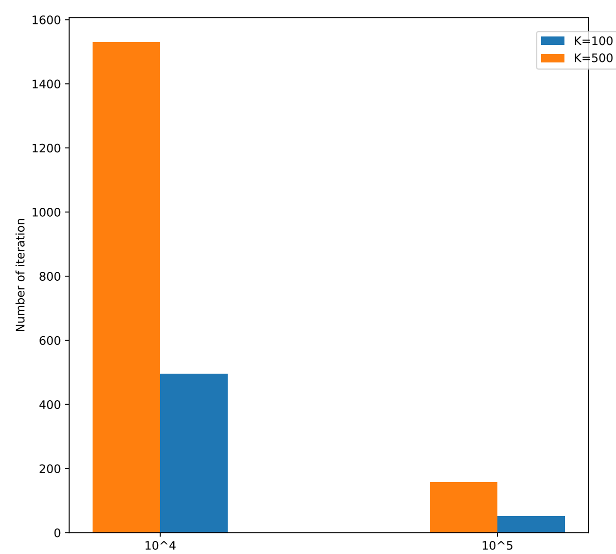

Fig. 2: Number of iteration in MINRES method

[1]All the materials employed for this paper are available at the project repository: https://github.com/ClaudioFantasia/Numerical-Optimization-for-large-scale-problems-and-Stochastic-Optimization.git