Exam

Numerical Methods and Optimization
University of Pisa, 2017-02-08

You may use Matlab, pencil or paper, or a calculator (unless explicitly stated in the exercise). You may use the quick reference sheet on Matlab's syntax posted on the web page of the course.

Exercise 1. Let $A \in \mathbb{R}^{m\times n}$ with full column rank, with $m > n$, and $b, c \in \mathbb{R}^m$. Consider the minimization problem

$$\min \; \|Ax - b\|^2 + \|Ax - c\|^2. \tag{1}$$

a. Show that the problem has the same solution as

$$\min \|Mx - v\|^2, \quad M = \begin{bmatrix} A \\ A \end{bmatrix}, \quad v = \begin{bmatrix} b \\ c \end{bmatrix}. \tag{2}$$

b. Show that the singular values of M are $\sqrt{2}\sigma_1, \sqrt{2}\sigma_2, \dots, \sqrt{2}\sigma_n$, where $\sigma_1, \sigma_2, \dots, \sigma_n$ are the singular values of A. *Hint:* what is the eigenvalue decomposition of M^*M?

c. Write a Matlab `function x = doublels_svd(A, b, c)` that solves the problem using the thin SVD (you may use Matlab's `svd(M, 0)` to compute it).

d. Report the computed output x and its absolute residual $\|Mx - v\|$, for the case

$$A = \begin{bmatrix} 10000 & 10000 \\ 10000 & 9999 \\ 10000 & 9999 \end{bmatrix}, \quad b = \begin{bmatrix} 1000 \\ 999 \\ 1000 \end{bmatrix}, \quad c = -b. \tag{3}$$

e. Write a Matlab `function x = doublels_normal(A, b, c)` that solves (1) using the method of normal equations on (2). Use the special structure of M to reduce the computational cost. What is the output of your program for A, b, c as given in (3)? Which of the two computed results is more accurate? Why?

f. What is the computational cost of `doublels_normal` for $m \gg n$? (It is sufficient to get the leading term correct, e.g., $3mn +$ lower order terms).

Exercise 2. Consider the following constrained optimization problem:

$$\begin{cases} \min\ x_1^2 + x_2^2 - 4x_2 \\ x_1^2 - x_2 \leq 0 \\ x_2 - 1 \leq 0 \end{cases}$$

a. Do global optimal solutions exist? Why?

b. Is it a convex problem? Why?

c. Do constraint qualifications hold in any feasible point?

d. Is the point $(0, 0)$ a local minimum? Why?

e. Find all the solutions of the KKT system.

f. Find local minima and global minima.

g. Find the objective function and constraints of the Lagrangian dual problem.

h. Is $(0, 1)$ an optimal solution of the Lagrangian dual problem? Why?

Exercise 3. Consider the following constrained optimization problem:

$$\begin{cases} \min\ x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + x_1x_3 + x_2x_4 - 4x_1 - 3x_2 - 2x_3 - x_4 \\ x_1 + x_2 + x_3 + x_4 \leq 2 \\ x \geq 0 \end{cases}$$

a. Is it a convex problem? Why?

b. Do global minima exist? Why?

c. Is the global minimum unique? Why?

d. Solve the problem by means of the Frank-Wolfe method with exact line search starting from the point $(0, 0, 0, 0)$ and setting the tolerance equal to 10^{-3}. What is the global minimum? What is the optimal value? How many iterations are needed?

e. Solve the problem by means of the penalty method starting from $\varepsilon = 1$, setting $\tau = 0.5$ and the tolerance equal to 10^{-3}. What is the global minimum? How many iterations are needed? *[Use the Matlab function* `fminunc` *for solving the unconstrained problem at each iteration with the following options:*
`options = optimoptions('fminunc','GradObj','on','Algorithm',`
`'quasi-newton','Display','off')`*]*

f. Solve the problem by means of the logarithmic barrier method starting from $\varepsilon = 1$, setting $\tau = 0.5$ and the tolerance equal to 10^{-3}. What is the global minimum? How many iterations are needed? *[Use the Matlab function* `fminunc` *for solving the unconstrained problem at each iteration with the same above options]*

Solutions

Exercise 1.

a. The two objective functions are equal: indeed, for every $u, w \in \mathbb{C}^n$, we have

$$\left\| \begin{bmatrix} u \\ w \end{bmatrix} \right\|^2 = \sum_{i=1}^{n} |u_i|^2 + \sum_{i=1}^{n} |w_i|^2 = \|u\|^2 + \|w\|^2.$$

So

$$\|Mx - v\|^2 = \left\| \begin{bmatrix} Ax - b \\ Ax - c \end{bmatrix} \right\|^2 = \|Ax - b\|^2 + \|Ax - c\|^2.$$

Note the two problems $\min\|Ax - b\|$ and $\min\|Ax - c\|$ in general have two different solutions, which do not coincide with the solution of (2). So an argument in that direction will not work.

b. If $A = USV^*$ is an SVD, then

$$M^*M = \begin{bmatrix} A^* & A^* \end{bmatrix} \begin{bmatrix} A \\ A \end{bmatrix} = A^*A + A^*A = 2A^*A = 2(VS^*U^*)(USV^*) = V(2S^*S)V^*$$

$$= V \begin{bmatrix} 2\sigma_1^2 & & & \\ & 2\sigma_2^2 & & \\ & & \vdots & \\ & & & 2\sigma_n^2 \end{bmatrix} V^*.$$

We have seen that for any matrix M the eigenvalues of M^*M are the squares of the singular values of M, so the singular values of M are $\sqrt{2}\sigma_1, \sqrt{2}\sigma_2, \ldots, \sqrt{2}\sigma_n$.

Note One can also prove that the SVD of M is

$$M = \begin{bmatrix} \frac{1}{\sqrt{2}}U & -\frac{1}{\sqrt{2}}U \\ \frac{1}{\sqrt{2}}U & \frac{1}{\sqrt{2}}U \end{bmatrix} \begin{bmatrix} \sqrt{2}S \\ 0 \end{bmatrix} V. \tag{4}$$

c.
```
function x = doublels_svd(A, b, c)
M = [A; A];
v = [b; c];
[U, S, V] = svd(M, 0);
d = U' * v;
% the following line can be replaced by y = S \ d, or a for cycle.
y = d ./ diag(S);
x = V * y;
```

```
>> A = [10000 10000; 10000 9999; 10000 9999];
>> b = [1000; 999; 1000];
>> x = doublels_svd(A, b, -b)
x =
   1.0e-09 *
   -0.3932
    0.3932
>> M = [A; A]; v = [b; -b];
>> norm(M*x - v)
ans =
   2.4487e+03
```

Different implementations may give slightly different results.

d.
```
function x = doublels_normal(A, b, c)
M_star_M = 2 * (A'*A);
M_star_v = A' * (b+c);
x = M_star_M \ M_star_v;
```

Note Computing M^*M as $2A^*A$ is the main optimization with respect to the case of general M, since it allows us to replace a $(n \times 2m) \cdot (2m \times n)$ matrix product with a $(n \times m) \cdot (m \times n)$ one. Similarly, we may compute

$$\begin{bmatrix} A^* & A^* \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} = A^*b + A^*c = A^*(b + c). \quad (5)$$

Another (very minor) optimization would have been computing the result using the formula `x = (A'*A) \ (A' * (b+c)/2)`, trading the scalar-matrix multiplication for a scalar-vector multiplication.

For the given A, b, c, the program produces

```
>> doublels_normal(A, b, -b)
ans =
     0
     0
```

Indeed, that 0 is the exact solution to the given problem whenever $c = -b$; this is evident from the formula (5) for the right-hand side, because $b + c = 0$. So the method of normal equations this time is more accurate.

Note All the possible variants of the method of normal equations that I have tried give the zero vector (exactly) as a solution. If your method does not produce the zero vector, I suspect there is something wrong with it; for instance, the parenthesization $((M^*M)^{-1}M^*)b$ that results in a significantly more expensive method.

f. In our implementation, the first line contains the product A^*A, which is $(n \times m) \cdot (m \times n)$ and costs $2mn^2$ operations, and the multiplication by 2, which costs $O(n^2)$. The second line contains $b + c$, which is $O(m)$, and the product $A^*(b + c)$, which costs $O(mn)$. The third line is the solution of a $n \times n$ linear system, which costs $O(n^3)$. The leading term for $m \gg n$ is $2mn^2$.

Note All implementations that use the optimization $M^*M = 2A^*A$ should give $2mn^2$ as an answer. All those that don't use it should give $4mn^2$ (cost of computing M^*M directly).

Exercise 2.

a. Yes, the objective function is continuous and the feasible region is closed and bounded (Weierstrass Theorem).

b. Yes, objective function and constraints are convex.

c. Yes, Slater's condition holds.

d. No, it does not solve the KKT system.

e. The KKT system has a unique solution $x = (0, 1)$ with $\lambda = (0, 2)$.

f. $(0, 1)$ is the global minimum.

g. The Lagrangian dual problem is

$$\begin{cases} \max & -\dfrac{(4 + \lambda_1 - \lambda_2)^2}{4} - \lambda_2 \\ \lambda_1 \geq 0 \\ \lambda_2 \geq 0 \end{cases}$$

h. No, the optimal value of the Lagrangian dual problem is -3, while the value of the dual objective function at $(0, 1)$ is $-13/4 < -3$.

Exercise 3.

a. Yes, the hessian matrix of the objective function f is

$$Q = \begin{pmatrix} 2 & 0 & 1 & 0 \\ 0 & 4 & 0 & 1 \\ 1 & 0 & 6 & 0 \\ 0 & 1 & 0 & 8 \end{pmatrix}$$

The eigenvalues of Q are 1.7639, 3.7639, 6.2361, 8.2361 hence f is strongly convex.

b. Yes, since f is strongly convex.

c. Yes, since f is strongly convex.

d. After 2 iterations the algorithm finds the approximated global minimum $x = (1.5000, 0.5000, 0.0000, 0.0000)$ with value -4.75.

e. After 9 iterations the algorithm finds the approximated global minimum $x = (1.5013, 0.5006, -0.0005, -0.0005)$.

f. After 13 iterations the algorithm finds the approximated global minimum $x = (1.4996, 0.4998, 0.0002, 0.0002)$.