Third Pset - Optimization Course, ECE TUC

Leonidas Bakopoulos 2018030036

December 14, 2022

Introduction

The purpose of this problem set was for the student to familiarize with projection optimization problems, in both theoretical form (by solving the dual using the KKT) and practical form (by implementing the projection gradient descent algorithm). It is worth mentioning that in order to implement the "project gradient descent" algorithm, the theoretical form is used in order to perform the projection step.

First Exercise

a)

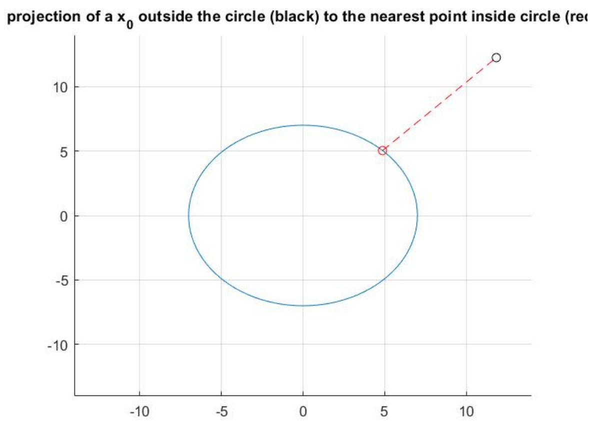


Figure 1: Projection problem in B(0,r)

In the figure above, is shown a black point outside the euclidean ball and its projection inside the ball (red point). The red line corresponds to the **minimum** distance from the point outside the circle to the ball. That leads us to the fact, that the red point, is the minimum projection. In case that a point already belongs to the circle, the minimum projection is the point itself.

Note: The figure was created using the calculations that are going to be explained in the following parts of the exercise.

b)

$$\boxed{\begin{array}{l}\text{minimize } \frac{1}{2}\left\|x - x_0\right\|_2^2 \\ \\ \text{subject to: } \left\|x\right\|_2^2 \leq r^2\end{array}}$$

Note: This constraint is equivalent to the $\|x\|_2 \leq r$, but the gradient of the first constraint is easier to compute.

c)

Dual is: $L(x, v, \lambda) = \frac{1}{2}\left\|x - x_0\right\|_2^2 + \lambda(\|x\|_2^2 - r^2)$
and the **KKT** are:

$\frac{\theta L}{\theta \vec{x}}\Big|_{x=x_*} = 0 \Rightarrow$
$x_* - x_0 + 2\lambda x_0 = 0$ (1)
$\|x_*\|_2^2 \leq r^2$ (2)
$\lambda \geq 0$ (3)
$\lambda(\|x_*\|_2^2 - r_2) = 0$ (4)

d)

In case of $\lambda > 0 \Rightarrow$
from (4) $\|x_*\|_2^2 = r^2 \xrightarrow{r>0} \|x_*\|_2 = r$
(so x_* is boundary of B(0,r)).

From (1) $x_*(1 + 2\lambda) = x_0 \Rightarrow x_* = \frac{x_0}{1+2\lambda}$
and $\|x_*\| = r \Rightarrow \frac{1}{1+2\lambda}\left\|x_0\right\| = r \Rightarrow$
$\lambda = \frac{1}{2r}\left\|x_0\right\| - \frac{1}{2}$

e)

if $\lambda = 0$ then from (1) $x_* = x_0$, so x_0 already belongs in the ball.

Second Exercise

a)

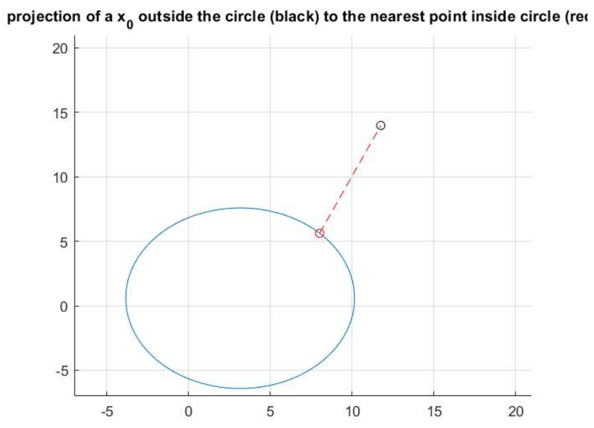


Figure 2: Projection problem in B(center,r)

b)

$$\boxed{\begin{array}{l}\text{minimize } \frac{1}{2}\left\|x - y - x_0\right\|_2^2 \\ \\ \text{subject to: } \|x - y\|_2^2 \leq r^2\end{array}}$$

c)

Dual is: $L(x, v, \lambda) = \frac{1}{2} \|x - y - x_0\|_2^2 + \lambda(\|x - y\|_2^2 - r^2)$
and the **KKT** are:

$\left.\frac{\theta L}{\theta \vec{x}}\right|_{x=x_*} = 0 \Rightarrow$
$x_* - y - x_0 + 2\lambda x_0 = 0 \Rightarrow$
$x_* = y + \frac{x_0}{2\lambda+1}$ (1)
$\|x_* - y\|_2^2 \leq r^2$ (2)
$\lambda \geq 0$ (3)
$\lambda(\|x_* - y\|_2^2 - r_2) = 0$ (4)

Note: We can see that the difference between this questions'(1) and the (1) of the previous exercise, is the center of the circle.

d)

Let $\lambda > 0$
Replacing x_* in (4) with the (1):

$\lambda(\|x_* - y\|_2^2 - r^2) = 0 \Rightarrow$

$\|x_* - y\|_2^2 = r^2 \Rightarrow$

$\left\| y + \frac{x_0}{2\lambda+1} - y \right\|_2^2 = r^2 \Rightarrow$

$\left\| \frac{x_0}{2\lambda+1} \right\|_2 = r \Rightarrow$

$\lambda = \frac{\|x_0\|}{2r} - \frac{1}{2}$

Note: That is the same as the first exercise

e)

if $\lambda = 0$ then from (1) $x_* = y + x_0$, so x_0 already belongs in the ball.

Third exercise

a)

In the figures below (same figure from a different angle) are depicted, the black point (outside the hyperplane $x \geq a$) and its projection on the hyperplane (red point).

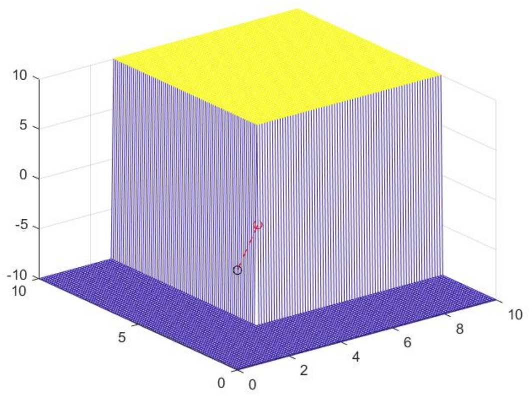


Figure 3: Projection problem in $x \geq a$

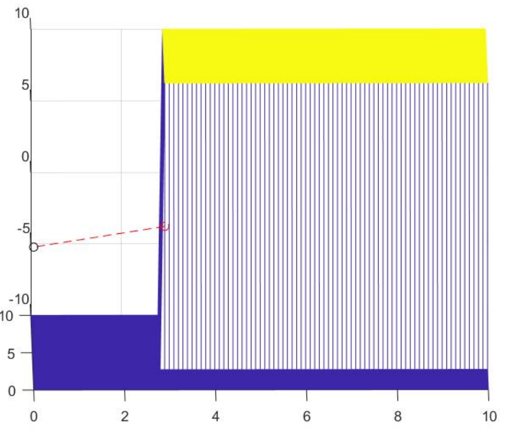


Figure 4: Projection problem in $x \geq a$

Theoretical solution

$$\boxed{\begin{aligned} &\text{minimize } \tfrac{1}{2}\left\|x - x_0\right\|_2^2 \\ &\text{subject to: } a - x \leq 0 \end{aligned}}$$

The Lagrangian is:
$L(x, v, \lambda) = \frac{1}{2}\left\|x - x_0\right\|_2^2 + \lambda(a - x)$

The **KKT** conditions:

$\left.\frac{\partial L}{\partial \vec{x}}\right|_{x=x_*} = 0 \Rightarrow$

$(x_* - x_0) = \lambda \Rightarrow$

$x_* = \lambda + x_0$ (1)

$x \geq \alpha$ (2)
$\lambda \geq 0$ (3)
$\lambda(a - x_*) = 0$ (4)

if $\lambda = 0$ then from (1), we can see that $x_* = x_0$ (same as previous, if $\lambda = 0$, x_0 already belongs in hyperplane).

If $\lambda > 0$ then $a - x_* = 0 \Rightarrow X_* = a$

```
1  a=5*rand(n,1);
2  ...
3  mesh(x,y,func )
4
5  %projection
6  hold on;
7  x_start=zeros(n,1);
8  x_projected=a;
9
10 z=zeros(dt,1);
11 plot3(x_1, y_1,z, '--r')
12 plot3(x_start(1),x_start(2),0,'ok')
13 plot3(x_projected(1),x_projected(2), 0, 'or')
```

Figures 3,4 were drawn with the code presented above. First of all, the creation of the hyperplane $x \geq \alpha$, was based on the following algorithm:

```
if($x \geq \alpha$) then function(x)= 10;
else function(x)=-10;
```

Then the point [0,0] was projected (line 8) and plotted using lines (lines 11,12,13).

Fourth Exercise

a)

minimize $\frac{1}{2}\left\|x\right\|_2^2$
subject to $a^\top x = b$

The **KKT** are:

$$\left.\frac{\partial L}{\partial \vec{x}}\right|_{x=x_*} = 0 \Rightarrow$$

$$x_* + v\alpha = 0 \Rightarrow$$

$$x_* = -v\alpha \ (1)$$

$$a^\top x_* = b \ (2)$$

multiplying (1) from left with $a^\top$: $a^\top x_* + a^\top v a = 0 \xrightarrow[b \epsilon R]{(2)} b = v\left\|a\right\|_2^2 \Rightarrow v = \frac{b}{\left\|a\right\|_2^2}$

so $x_* = -\frac{ba}{\left\|a\right\|_2^2}$

b)

In this subsection,is going to be explained the project gradient descent algorithm (PGD) .The PGD, consists of two basic steps: [1]

1. calculate the gradient descent
2. project step ones' point, on the constraint hyperplane

In order to solve the projection step, I have to solve the:

$$\boxed{\begin{array}{l}\text{argmin } \frac{1}{2}\left\|x-x_0\right\|_2^2 \\ \text{subject to } a^\top = b\end{array}}$$

where x_0 is the point that was calculated during the Gradient Descent step.

KKT

$x_* - x_0 + ua = 0$ (1)
$a^\top x_* = b$ (2)

using the same steps as the first question, we can see that:

$b - a^\top x_0 + a^\top va = 0 \xrightarrow{v\epsilon R}$
$b - a^\top x_0 + v\left\|a\right\|_2^2 = 0 \Rightarrow$
$v = \frac{a^\top x_0 - b}{\left\|a\right\|_2^2}\epsilon R$

and

$x_* = x_0 - va =$
$= x_0 - \frac{a^\top x_0 - b}{\left\|a\right\|_2^2}a$

Conclusion on PGD algorithm:

```
1.start from a point
2. while(!termination)
3.$x_{k+\frac{1}{2}} = x_{k*} - \frac{1}{L}\nabla f_0(x_k)$
4.$x_{k+1} = x_{k+\frac{1}{2}} - \frac{a^\top x_{k+\frac{1}{2}} - b}{\left\|a\right\|_2^2}$
5.$x_k = x_{k+1}$
6. endwhile
```

Visualized results for n=2

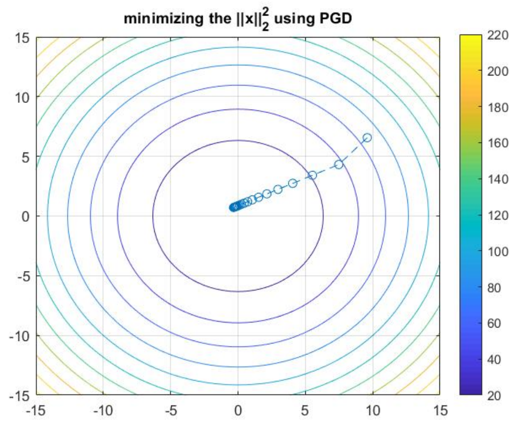
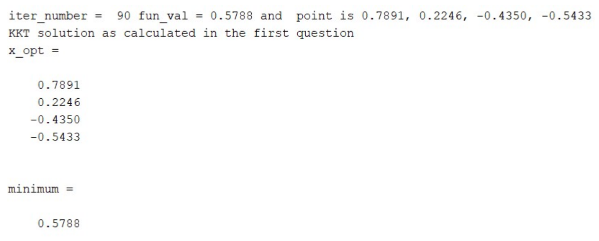


Figure 5: Minimization process



```
iter_number =  90 fun_val = 0.5788 and  point is 0.7891, 0.2246, -0.4350, -0.5433
KKT solution as calculated in the first question
x_opt =

    0.7891
    0.2246
   -0.4350
   -0.5433


minimum =

    0.5788
```

Figure 6: command line output

Note: the starting point of the above minimization is given by $x = 10 * rand(n,1)$; As we can see in the figures above, the optimal point is not [0,0] but one close to it (as it was expected cause [0,0] does not satisfy the constraint).
Also, as it is shown in figure 7, the theoretical and the practical results are the same.Finally, I would like to note that as termination condition,was used $\frac{x_{k+1}-x_k}{\|x_{k+1}\|} < \varepsilon$ and not $\frac{x_{k+1}-x_k}{\|x_k\|} < \varepsilon$. In the second case, it was observed that the algorithm needs more iterations to converge (in case of $n > 4$).
Code for ex. 4

```
%PGD algorithm

x_k=x_start;

while(true)
   x_temp=x_k-(1/L)*g(x_k); %gradient step
   x_k_plus=x_temp-((a'*x_temp-b)/norm(a)^2)*a; %projection step
   if(norm((x_k_plus-x_k))/norm(x_k_plus) < epsilon)
       break;
   end
   x_k=x_k_plus;
end
```

Fifth exercise

a)

minimize $\|x - x_0\|_2^2$
subject to $Ax = b$

The Lagrangian of the problem above is:
$L(x,v) = \frac{1}{2}\|x - x_0\|_2^2 + v(A^\top x - b)$
KKT:

$$\left.\frac{\partial L}{\partial \vec{x}}\right|_{x=x_*} = 0 \Rightarrow$$
$x_* = -(vA^\top + x_0)$ (1)

$Ax_* - b = 0 \Rightarrow x_* = A^{-1}b$ (2)

Replacing (1) with (2):
$A^{-1}b = -vA^\top - x_0 \Rightarrow$
$v = -A^{-1}b(A^\top)^{-1} - x_0(A^\top)^{-1}$

b)

minimize $\frac{1}{2}x^\top Px + q^\top x$
subject to $Ax = b$

The Lagrangian of the problem above is:
$L(x,v) = \frac{1}{2}x^\top Px + q^\top x + v(A^\top x - b)$
KKT:

$$\left.\frac{\partial L}{\partial \vec{x}}\right|_{x=x_*} = 0 \Rightarrow$$
$Px_* + q^\top + A^\top v = 0$ (1)

$Ax_* = b$ (2)

So the above 2x2 system of equations,is equivalent to the matrix equation below:
$\begin{bmatrix} P & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} x_* \\ v \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$ (3)

b)

The first step in order to solve the optimization problem, was to create the matrices A and P. P matrix is a positive definite (n by n) matrix and was constructed as it was also done in the second pset of this course. A matrix is a p by n matrix where rank(A)= p. In order to satisfy this constraint,is used the randn(p,n) function in matlab where p¡n.

```matlab
%Creation of P
n=4;
A=rand(n,n); %Note A has nothing to do with the constraint matrix
[U,S,V]=svd(A);

l_min=1; %we choose postive numbers cause we want
K=3;
l_max=K*l_min;
z = l_min + (l_max - l_min) * rand(n - 2, 1);
eig_P=[l_min; l_max; z];
Lamda=diag(eig_P);


P=U*Lamda*U'; %P is positive definite
q=((-1).^(randi([0,1],n,1))).*rand(n,1);
```

```matlab
%%creating matrices for the costraint

n=4;
p=n-1; %using p<n in order to create a full rank A matrix
A=randn(p,n); %creates a p,n table with rank p
b=((-1).^(randi([0,1],p,1))).*rand(p,1);
```

ii)

In order to solve the problem using the KKT conditions, I have to solve equation (3). This was accomplished using the linsolve() function.

```matlab
function [ret_val] = KKT_solution(A,P,q,b)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % All A,P,q,b are generated     %
    % "randomly" in the previous    %
    % qusestions                    %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [¬,s_2]=size(A');
    z=zeros(s_2);
    matrix=[P,A';A,z];
    equals=[-q;b];
    solution=linsolve(matrix,equals);
    x_opt=solution(1:4);
    ret_val=f(x_opt,P,q);
    fprintf('KKT solution is %2.5f\n', ret_val);
end
```

iii)

Last but not least, a same as previous PGD algorithm is used in order to calculate the optimal value of the quadratic function. The only thin that changes is the project step, that was adapted in this problem (and the above calculations).

```matlab
1  function [x,y,z] = PGD(x_start,L,P,q,A,b,epsilon)
2  x_k=x_start;
3  while(true)
4      x_temp=x_k-(1/L)*g(x_k,P,q); %gradient step
5      v=pinv(A')*(pinv(A)*(A*x_temp-b));
6      x_k_plus=x_temp-A'*v; %projection step
7      if(norm((x_k_plus-x_k))/norm(x_k_plus) < epsilon)
8          break;
9      end
10     x_k=x_k_plus;
11 end
```

In the projection step, it was needed to project the $x_{k+\frac{1}{2}}$ to the hyperplane $Ax = b$. This problem was solved in the first question and the result was used in line 6.

Visualized results for n=2

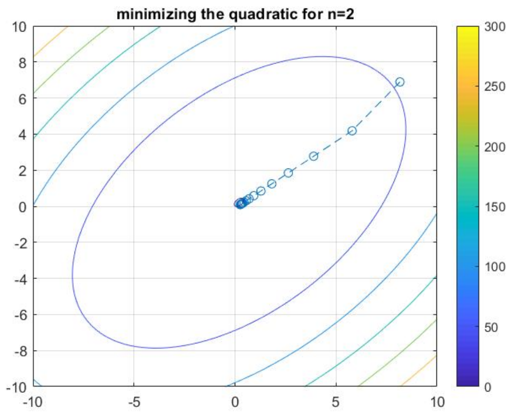
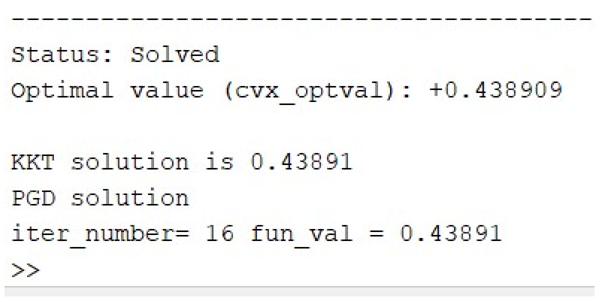


Figure 7: Minimization process



```
----------------------------------------
Status: Solved
Optimal value (cvx_optval): +0.438909

KKT solution is 0.43891
PGD solution
iter_number= 16 fun_val = 0.43891
>>
```

Figure 8: command line output for n=4

As is shown in figure 10, both KKT and PGD are calculating the correct minimum (based on the CVX method). It is worth mentioning that the PGD converges faster in exercise 5 than exercise 4.

Submitted code

The code that is attached with this this report, implements the solution for exercise 4 and 5. This code is initialized with n=4. In order to ”recreate” the graphs you must set n=2 and re run the main script.

References

[1] Projected Gradient Descent explained (Columbia)