RO – TP 4 : KKT et dualité

Stéphane Canu

Décembre 2016, ASI, INSA Rouen

Le but du TP est de calculer les conditions KKT et la dualité.

Ex. 1 — programmation linéaire et quadratique : dualité et KKT

1. programmation linéaire, dualité et KKT
   a) résoudre le programme linéaire suivant :

$$\begin{cases} \min\limits_{\mathrm{x}\in\mathbf{R}^m} & \mathrm{f}^\top \mathrm{x} \\ \text{avec} & A\mathrm{x} \geq \mathrm{b} \\ \text{et} & \mathrm{x} \geq 0 \end{cases}$$

avec

$$f = \begin{pmatrix} 30 \\ 20 \end{pmatrix} \qquad b = \begin{pmatrix} 24 \\ 36 \\ 4 \\ -5 \end{pmatrix} \qquad A = \begin{pmatrix} 8 & 12 \\ 12 & 12 \\ 2 & 1 \\ -1 & -1 \end{pmatrix}$$

```
f= [30 ; 20];
b = [24;36;4;-5];
A = [8 12 ; 12 12 ; 2 1 ; -1 -1];
cvx_begin
   variable x(2)
   dual variables a p;
   minimize( f'*x )
   subject to
       a : A*x >= b
       p : x >= 0;
cvx_end
```

   b) Ecrire le lagrangien et vérifiez l'optimalité de la solution (les KKT). Qu'en pensez vous ?

```
f - A'*a - p
sign(A*x-b)
sign(x)
sign(a)
sign(p)
a.*(A*x-b)
p.*x
```

   c) en déduire une manière de calculer x connaissant a

```
ind = find(abs(a)>0.00001)
A(ind,:)\b(ind)
```

   d) Ecrire le programme dual et le résoudre en CVX

```
cvx_begin
   variable y(4)
   dual variables d e;
   maximize( b'*y )
   subject to
       d : A'*y <= f;
       e : y >=0 ;
cvx_end
```

   e) Résoudre le problème primal à l'aide de `linprog` (ou `GLPK` ou un autre solveur)

```
[xl c e o du] = linprog(f,-A,-b,[],[],0*f);
```

f) Vérifiez l'optimalité de la solution

```
a = du.ineqlin;
p = du.lower;
f - A'*a - p
sign(A*xl-b)
sign(xl)
sign(a)
sign(p)
a.*(A*xl-b)
p.*x
```

2. Nous allons maintenant étudier deux exemples de programmes quadratiques

a) résoudre le problème suivant :

$$\begin{cases} \min\limits_{x_1,x_2} & x_1^2 + x_2^2 + \frac{1}{2}x_1x_2 - 6x_1 - 8x_2 \\ \text{avec} & x_1 + x_2 = 5 \\ \text{et} & x_1 \geq 0; x_2 \geq 0 \end{cases}$$

```
cvx_begin
   variables x1 x2
   dual variables d e1 e2;
   minimize( .75* x1^2 + .75*x2^2 +.25*(x1+x2)^2 - 6* x1 - 8 *x2 )
   subject to
       d : x1 + x2 == 5;
       e1 : x1 >=0 ;
       e2 : x2 >=0 ;
cvx_end
```

voir page 35 du *CVX Users' Guide, Release 2.1.*

b) résoudre ce même problème sous forme matricielle

$$\begin{cases} \min\limits_{\mathrm{x}\in\mathbb{R}^2} & \frac{1}{2}\mathrm{x}^\top Q\mathrm{x} + f^\top \mathrm{x} \\ \text{avec} & A\mathrm{x} = b \\ \text{et} & \mathrm{x} \geq 0 \end{cases}$$

avec Q une matrice, f, A des vecteurs et b un scalaire que l'on précisera

```
Q = 2*[1 0.25 ; 0.25 1];
f = [-6 ; -8];
A = [-1 -1];
b = -5;
tic
cvx_begin
   variables x(2)
   dual variables d e;
   minimize( .5*x'*Q*x + f'*x )
   subject to
       d : -A*x == -b;
       e : x >=0 ;
cvx_end
```

c) vérifiez que le problème est bien convexe

```
eig(Q)
```

d) Ecrire le lagrangien et vérifiez l'optimalité de la solution (les KKT). Qu'en pensez vous ?

```
Q*x + f + A'*d - e
A*x-b
sign(e)
e.*x
```

e) A partir des KKT, recalculez la solution optimale et vérifiez que l'on obtient bien la même solution que CVX

```
[Q A'; A 0]\[-f;b]
```

f) On se propose maintenant de résoudre un programme quadratique avec

$$Q = \begin{pmatrix} 1 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 1 \end{pmatrix} \qquad f = \begin{pmatrix} -6 \\ -8 \\ -2 \end{pmatrix} \qquad b = -5 \qquad A = \begin{pmatrix} -1 & -1 & -1 \end{pmatrix}$$

```
Q = 2*[1 0.25 0 ; 0.25 1 .25 ; 0 .25 1];
f = [-6 ; -8 ; -2];
A = [-1 -1 -1];
b = -5;

cvx_begin
   variables x(3)
   dual variables d e;
   minimize( .5*x'*Q*x + f'*x )
   subject to
        d : A*x == b;
        e : x >=0 ;
cvx_end
```

g) vérifiez l'optimalité de la solution (les KKT). Qu'en pensez vous ?

```
Q*x + f - A'*d - e
-A*x+b
sign(e)
e.*x
```

h) recalculez la solution à partir de la connaissance des ensembles I_x et I_0

```
ind = find(abs(e)<0.00001)
sol = [Q(ind,ind) -A(ind)'; -A(ind) 0]\[-f(ind);-b];
xsol = zeros(3,1);
xsol(ind) = sol(1:length(ind));
[x xsol]
```

i) vérifiez l'optimalité de la solution (les KKT). Qu'en pensez vous ?

```
esol = e;
esol(ind) = 0;

Q*xsol + f - A'*sol(end) - esol
A*xsol-b
sign(esol)
esol.*x
```

j) résoudre la même problème à l'aide de `quadprog`

```
xq = quadprog(Q,f,[],[],A,b,zeros(3,1));
```

k) vérifiez que les solutions sont bien le mêmes (ou presque)

```
[x xsol xq]
```

3. Ecrire une fonction `KKT_check`$(Q, f, A, b, x, \lambda, \mu)$ qui vérifie que x est bien une solution optimale d'un programme quadratique standard associée aux multiplicateurs de Lagrange λ et μ.

$$\begin{cases} \min\limits_{\mathrm{x}\in\mathbb{R}^n} & \frac{1}{2}\mathrm{x}^\top Q\mathrm{x} + f^\top \mathrm{x} \\ \text{avec} & A\mathrm{x} = b \\ \text{et} & \mathrm{x} \geq 0. \end{cases}$$

Bonus : comment faudrait-il faire pour vérifier les KKT de ce problème si les multiplicateurs de Lagrange λ et μ n'étaient pas connus ?