Karush Kuhn-Tucker Conditions

by sumihar christian 0611154000115

Lagrange Multiplication

$$minf(x)$$

constraint to

$$h(x) = 0g(x) <= 0$$

Where Karush Kuhn-Tucker Conditions is:

$$L = (x, \lambda) = f(x) - \lambda g(x)$$

Where the x^* is a minimum and regular point respect to the constraint $\exists \lambda \in \mathbb{R}$, $\mu \in \mathbb{R}$, where $\mu >= 0$ and

$$\nabla(x^*) + \lambda^T \times \nabla(x^*) + \mu^T \times \nabla g(x^*) = 0$$

$$\mu^T \times g(x^*) = 0$$

where that is the final version of the first order necessary conditions of a problem which has both equality or inequality constraint

KKT necessary conditions

$$f(x) \dots (1)$$

$$g(x) - b_i >= 0, i = 1, \dots, k \dots (2)$$

$$g(x) - b_i = 0, i = 1, \dots, k \dots (3)$$

Description	Equation	Applies To
Feasibility	$g_i(x^*) - b_i$	2,3
No direction which improves objective and is feasible	$\nabla f(x^*) - \sum_{i=1}^{m} \lambda_i^* g_i(x^*) = 0$	1,2,3
Complementary slackness	$\lambda^*(g_i(x^*) - b_i) = 0, i = 1, \dots, k$	2
Positive Lagrange multipliers	$\lambda_i^* >= 0, i = 1, \dots, k$	2

Problem Defined

given

$$f(x) = (x_1 - 8)^2 + (x_2 - 8)^2$$

constraint to

$$g_1(x) = x_1 + x_2 <= 12$$

$$g_2(x) = x_1 <= 6$$

$$g_3(x) = x_1, x_2 >= 0$$

check if is feasible

$$g_i(x^*) - b_i$$

$$x_1 + x_2 - 12 <= 0$$

$$x_1 - 6 <= 0$$

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} - \lambda_1 \begin{bmatrix} \frac{\partial g_1}{\partial x_1} \\ \frac{\partial g_1}{\partial x_2} \end{bmatrix} - \lambda_2 \begin{bmatrix} \frac{\partial g_2}{\partial x_1} \\ \frac{\partial g_2}{\partial x_2} \end{bmatrix} = 0$$

Solution

matrix

In [16]:

```
# define libraries
import numpy as np
import matplotlib.pyplot as plt
```

$$(x_1 - 8)^2 + (x_2 - 6)^2$$

constraint to

$$g(x) = x_1 + x_2 - 18 >= 0$$

In [23]:

```
A = np.matrix('8,0,3,2;0,4,1,4;3,1,0,0;2,4,0,0')
b = np.matrix('0;0;8;15')
solution = np.array(np.linalg.inv(A)*b)
r = np.sqrt(4*solution[0]**2+2*solution[1]**2)
print(solution)
print(r)
```

```
[[ 1.7 ]
 [ 2.9 ]
 [-3.12]
 [-2.12]]
[5.32728824]
```

solution for $x_1, x_2, \mu_1, \mu_2,\$ [[1.7] [2.9] [-3.12] [-2.12]]

visualization

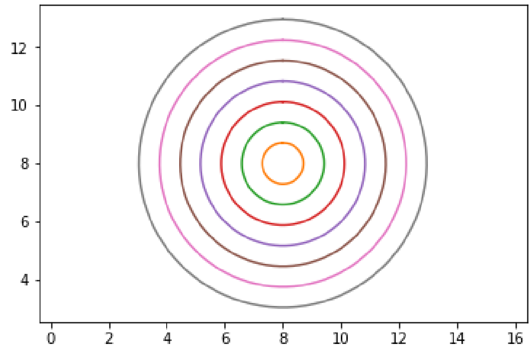
In [14]:

```
#plot objective function
x = 8*np.ones(8)
y = 8*np.ones(8)
r = np.arange(8)/np.sqrt(2)

phi = np.linspace(0.0,2*np.pi,100)
na = np.newaxis
```

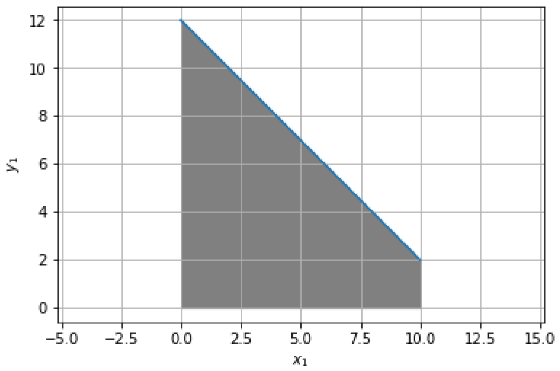
In [15]:

```
x_line = x[na,:]+r[na,:]*np.sin(phi[:,na])
y_line = y[na,:]+r[na,:]*np.cos(phi[:,na])
plt.axis('equal')
ax = plt.plot(x_line,y_line,'-')
```



In [12]:

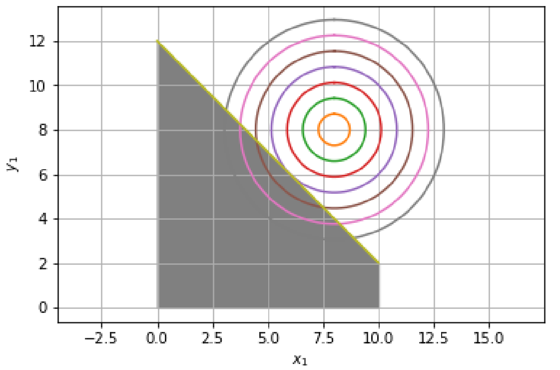
```
#plot the lines
x1 = np.linspace(0,10,1000)
x2 = 12*np.ones(1000)-x1
bx = plt.plot(x1,x2,label='$x_1+x_2-12<=0$')
d= np.zeros(len(x2))
plt.fill_between(x1,x2, where=x2>=d,color='grey')
plt.axis('equal')
plt.grid()
plt.xlabel('$x_1$')
plt.ylabel('$y_1$')
plt.show()
```



In [11]:

```
x_line = x[na,:]+r[na,:]*np.sin(phi[:,na])
y_line = y[na,:]+r[na,:]*np.cos(phi[:,na])
ax = plt.plot(x_line,y_line,'-')

bx = plt.plot(x1,x2,label='$x_1+x_2-12<=0$')
d= np.zeros(len(x2))
plt.fill_between(x1,x2, where=x2>=d,color='grey')
plt.axis('equal')
plt.grid()
plt.xlabel('$x_1$')
plt.ylabel('$y_1$')
plt.show()
```



summary of conditions

Cost	Constraint	λ
$f(x)$	$g(x) = 0$	< 0
$f(x)$	$g(x) >= 0$	0
$f(x)$	$g(x) <= 0$	< 0

thank you, svmihar 2019 (github.com/svmihar)
made with ~~matlab~~ ~~latex~~ jupyternotebook with python 3.7