COMPSCI 671 Srping 2019

HW 4

Yiteng Lu

1. Convexity I

1.1 Answer:

Suppose function f and g are convex functions, thus we have:

$$f\Big(\theta\mathbf{x}+(1-\theta)\mathbf{z}\Big) \leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{z})$$
$$g\Big(\theta\mathbf{x}+(1-\theta)\mathbf{z}\Big) \leq \theta g(\mathbf{x}) + (1-\theta)g(\mathbf{z})$$

Suppose $h(x) = f(x) + g(x)$, then we have:

$$\begin{aligned} h\Big(\theta\mathbf{x}+(1-\theta)\mathbf{z}\Big) &= f\Big(\theta\mathbf{x}+(1-\theta)\mathbf{z}\Big) + g\Big(\theta\mathbf{x}+(1-\theta)\mathbf{z}\Big) \\ &\leq \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{z}) + \theta g(\mathbf{x}) + (1-\theta)g(\mathbf{z}) \\ &\leq \theta(f(\mathbf{x}) + g(\mathbf{x})) + (1-\theta)(f(\mathbf{z}) + g(\mathbf{z})) \\ &\leq \theta h(\mathbf{x}) + (1-\theta)h(\mathbf{z}) \end{aligned}$$

$h\Big(\theta\mathbf{x}+(1-\theta)\mathbf{z}\Big) \leq \theta h(\mathbf{x}) + (1-\theta)h(\mathbf{z})$ fullfills the definition of a convex function, thus we can conclude that the sum of two convex functions is also convex.

1.2 Answer:

Given $f(x) =$, to prove f is convex, prove $f\Big(\theta x + (1-\theta)z\Big) \leq \theta f(x) + (1-\theta)f(z)$

$\cdot h$ is convex and g_i is convex and h is increasing in its i-th compoenent:

Proof:
Becase h is a convex function, we have:

$$h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (1-\theta)g_2(z), \ldots \theta g_n(x) + (1-\theta)g_n(z)) \leq \theta h(g_1(x), g_2(x), .., g_n(x)) + \\ \leq \theta f(x) + (1-\theta)f(z)$$

Becuase g_i is convex, we have:

$$g_n(\theta x + (1-\theta)z) \leq \theta g_n(x) + (1-\theta)g_n(z)$$

Because h is increasing in each ith compoenent, so we have:

$$h(g_1(\theta x + (1-\theta)z), g_2(g_n(\theta x + (1-\theta)z)), \ldots g_n(\theta x + (1-\theta)z)) \leq h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (\\ f\Big(\theta x + (1-\theta)z\Big) \leq$$

Therefore, accodring to the transitivity of inequalities:

$$f\Big(\theta x + (1-\theta)z\Big) \leq h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (1-\theta)g_2(z), \ldots \theta g_n(x) + (1-\theta)g_n(z)) \leq \theta f(x) +$$

$$f\Big(\theta x + (1-\theta)z\Big) \leq \theta f(x) + (1-\theta)f(z)$$

f is convex

$\cdot h$ is convex and g_i is affine functions:

Proof:
Becase h is a convex function, we have:

$$h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (1-\theta)g_2(z), \dots \theta g_n(x) + (1-\theta)g_n(z)) \leq \theta h(g_1(x), g_2(x), .., g_n(x)) - \\ \leq \theta f(x) + (1-\theta)f(z)$$

Becuase g_i is affine function, we have:

$$g_n(\theta x + (1-\theta)z) = \theta g_n(x) + (1-\theta)g_n(z)$$

Thus, so we have:

$$h(g_1(\theta x + (1-\theta)z), g_2(g_n(\theta x + (1-\theta)z)), \dots g_n(\theta x + (1-\theta)z)) = h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + ($$
$$f\Big(\theta x + (1-\theta)z\Big) =$$

Given two inequalities:

$$h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (1-\theta)g_2(z), \dots \theta g_n(x) + (1-\theta)g_n(z)) \leq \theta f(x) + (1-\theta)f(z)$$
$$f\Big(\theta x + (1-\theta)z\Big) = h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (1-\theta)g_2(z), \dots \theta g_n(x) + (1-\theta)g_n(z))$$

Therefore, accodring to the transitivity of equalities:

$$f\Big(\theta x + (1-\theta)z\Big) \leq \theta f(x) + (1-\theta)f(z)$$

f is convex

$\cdot h$ is convex and g_i is concave and h is decreasing in its i-th compoenent:

Proof:
Becase h is a convex function, we have:

$$h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (1-\theta)g_2(z), \dots \theta g_n(x) + (1-\theta)g_n(z)) \leq \theta h(g_1(x), g_2(x), .., g_n(x)) - \\ \leq \theta f(x) + (1-\theta)f(z)$$

Becuase g_i is concave, we have:

$$\theta g_n(x) + (1-\theta)g_n(z) \leq g_n(\theta x + (1-\theta)z)$$

Because h is decreasing in each ith compoenent, so we have:

$$h(g_1(\theta x + (1-\theta)z), g_2(g_n(\theta x + (1-\theta)z)), \dots g_n(\theta x + (1-\theta)z)) \leq h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (\\ f\Big(\theta x + (1-\theta)z\Big) \leq$$

Therefore, accodring to the transitivity of inequalities:

$$f\Big(\theta x + (1-\theta)z\Big) \leq h(\theta g_1(x) + (1-\theta)g_1(z), \theta g_2(x) + (1-\theta)g_2(z), \dots \theta g_n(x) + (1-\theta)g_n(z)) \leq \theta f(x) -$$

$$f\Big(\theta x + (1-\theta)z\Big) \leq \theta f(x) + (1-\theta)f(z)$$

f is convex

1.3 Answer:

Assume the maximum of the convex function f over the polyheron is achieved on a point x_0 which is inside the polyhedron,
meanwhile it exits two points on the boundary x_1 and x_2 which the connecting line of them passes through x_0 :

$$x_0 = \lambda x_1 + (1-\lambda)x_2 \tag{1}$$

therefore , due to the properties convexity:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2) \tag{2}$$

Given $f(x_0)$ is the maximum value:

$$\lambda f(x_1) + (1-\lambda)f(x_2) < \lambda f(x_0) + (1-\lambda)f(x_0) = f(x_0) \tag{3}$$

Use the transitive relation of inequalities from (2) , (3):

$$f(\lambda x_1 + (1-\lambda)x_2) < f(x_0) \tag{4}$$

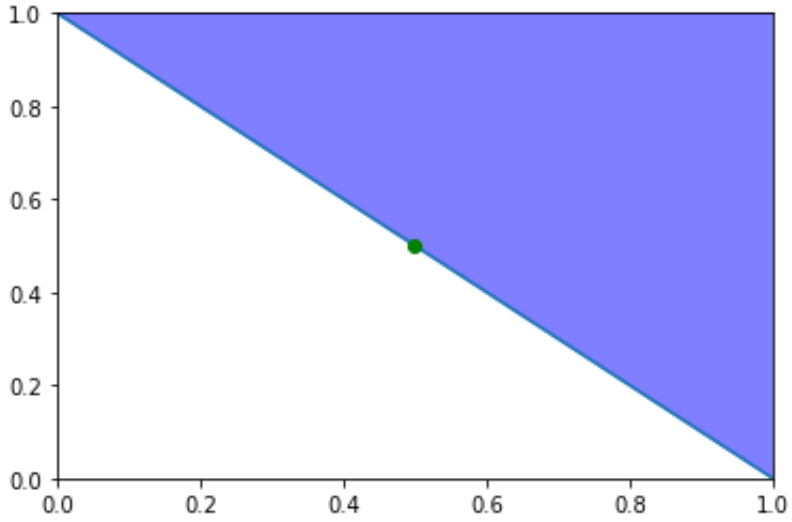
Inequality (4) shows that $f(\lambda x_1 + (1-\lambda)x_2) = f(x_0) < f(x_0)$ which rises the contradition.
Therefore, the maximum value cannot be achieved from the interior of the polyheron.

Assume the maximum value can be achieved on a point x_0 which lies on the edge of the polyhedron,
meanwhile we can find two points x_1 and x_2 (two vertices on that edege) whose connecting segment goes through x_0 ,
we can use the same proof to show it encounters the similar contradiction.
In conclusion, the maximum of a convex function f over the polyhedron P can only be found on one of its vertices.

2. Convexity II

2.1 Answer:

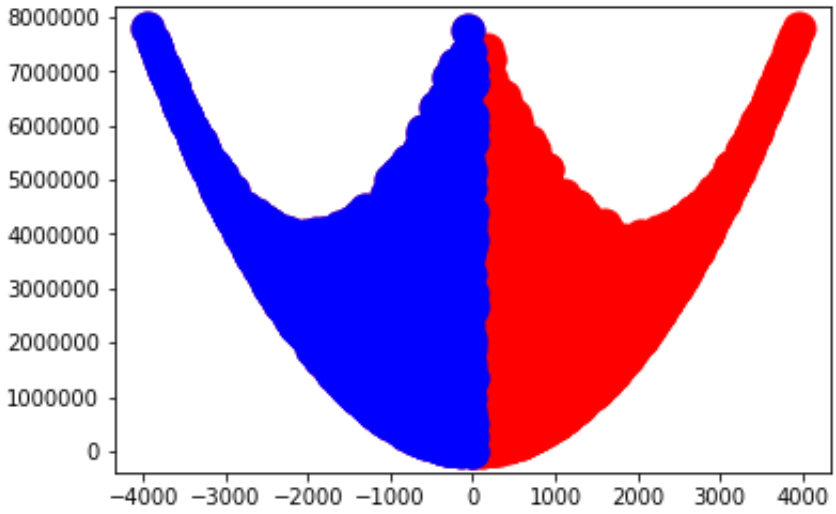
In [32]:
```
import matplotlib.pyplot as plt
import numpy as np

x1 = np.arange(-5,5,1)
x2 = 1- x1
plt.plot(x1,x2)
plt.xlim(0,1)
plt.ylim(0,1)
x1_star = 1/2
x2_star = 1/2
plt.plot(x1_star, x2_star,color ='green', marker='o')
y1 = 10
plt.fill_between(x1, x2, y1, color='blue', alpha='0.5')
plt.show()
```



2.2 Answer:

In [94]:
```
import random
w1 =random.sample(range(-2000, 2000), 4000)
w2 =random.sample(range(-2000,2000),4000)
g = np.ones(4000) - np.array(w1)-np.array(w2)
f = np.array(w1)**2+ np.array(w2)**2
index = np.where(g<=0)
f_fe  = f[index]
g_fe  = g[index]
```

In [96]:
```
plt.plot([g],[f],marker='o',markersize=15,color='red')
plt.plot([g_fe],[f_fe],marker='o',markersize=15,color = 'blue')
plt.show()
```



2.3 Answer:

Use the KKT conditions
Lagrangian stationary:

$$\frac{\partial \mathcal{L}(\mathbf{x},\lambda)}{\partial x_1} = \frac{d\, x_1^2+x_2^2+\lambda(1-x_1-x_2)}{dx_1} = 2x_1 - \lambda = 0$$

$$\frac{\partial \mathcal{L}(\mathbf{x},\lambda)}{\partial x_2} = \frac{d\, x_1^2+x_2^2+\lambda(1-x_1-x_2)}{dx_2} = 2x_2 - \lambda = 0$$

$$x_1 = \frac{\lambda}{2}$$

$$x_2 = \frac{\lambda}{2}$$

Complementary Slackness: $\lambda = 0$ or $g(\mathbf{x}) = 0$

if $\lambda = 0$, then the primal feasibility will not hold: $1 - 0 - 0 > 0$

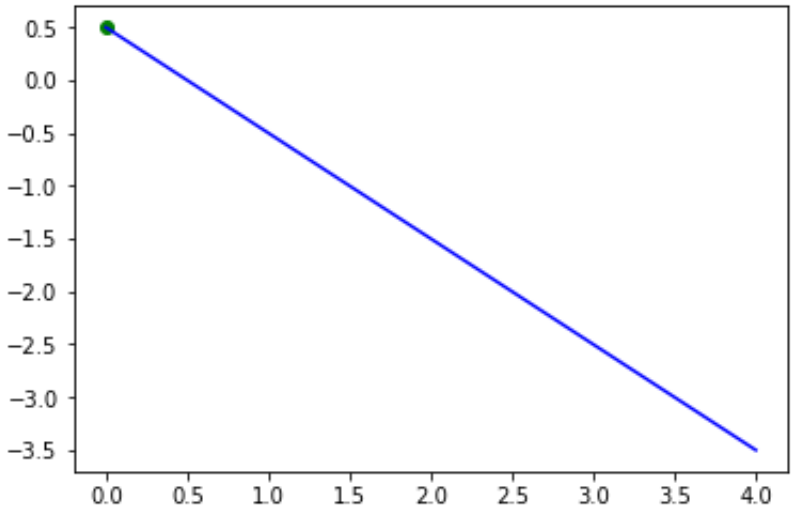
then we have: $g(\mathbf{x}) = 0, 1 - x_1 - x_2 = 0$

$$\frac{\lambda}{2} + \frac{\lambda}{2} = 1, \lambda = 1$$

$x_1 = 0.5, x_2 = 0.5$

$y^* = 0$
$z^* = 0.5$
$\lambda^* = 1$

In [93]:
```
plt.plot(0, 0.5,color ='green', marker='o')
y = np.arange(0,5)
z = 0.5 -y

plt.plot(y,z, color = 'blue')
plt.show()
```



2.4 Answer:

$q(\lambda) = \min_x [f(x) + \lambda g(x)]$

For any fixed value $\mathbf{x}$, the quantity inside the brackets is an affine function of λ, because $g(x)$ and $f(x)$ are just constants as far as only λ is concerned. Therefore, $q(\lambda)$ is concave

3. Support Vector Machine

3.1 Answer:

$$f(w, \epsilon_i) = \min_w \left(\frac{1}{2}\|w\|_2^2 + C \cdot \sum_i \epsilon_i \right)$$

subject to:

$$g([w, b], \epsilon) = (1 - \epsilon_i) - y_i(w^T x_i + b) \leq 0 \ \forall i \in \{1, \ldots, n\}$$

$$h(\epsilon) = -\epsilon_i \leq 0 \ \ \forall i \in \{1, \ldots, n\}$$

leading to the Lagrangian:

$$\mathcal{L}(w, b, \epsilon, a, r) = \frac{1}{2}\|w\|_2^2 + C \sum_i \epsilon_i - \sum_i \alpha_i \left[y_i(w^T x_i + b) - (1 - \epsilon_i) \right] - \sum_i r_i \epsilon_i$$

KKT conditions:

- Lagrangian stationarity:

$$\nabla_w \mathcal{L}(w, b, \epsilon, a, r) = w - \sum_i \alpha_i y_i x_i = 0 \quad \rightarrow w = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial}{\partial b} \mathcal{L}(w, b, \epsilon, a, r) = -\sum_i \alpha_i y_i = 0 \quad \rightarrow \sum_i \alpha_i y_i = 0$$

$$\frac{\partial}{\partial \epsilon} = C - \alpha_i - r_i = 0 \quad \rightarrow C = \alpha_i + r_i$$

- dual feasibility:

$$\alpha_i^* \geq 0$$
$$r_i^* \geq 0$$

- complementary slackness:

$$\alpha_i^* \left[(1 - \epsilon_i^*) - y_i(w^T x_i + b^*)\right] = 0$$
$$r_i \epsilon_i^* = 0$$

- primal feasibility:

$$g([w, b], \epsilon) = (1 - \epsilon_i) - y_i(w^T x_i + b) \leq 0 \ \forall i \in \{1, \ldots, n\}$$
$$h(\epsilon) = -\epsilon_i \leq 0 \ \ \forall i \in \{1, \ldots, n\}$$

dual objective:
$\mathcal{L}(w, b, \epsilon, a, r) = \frac{1}{2}\|w\|_2^2 + C \sum_i \epsilon_i - \sum_i \alpha_i y_i w^T x_i - \sum_i \alpha_i y_i b + \sum_i \alpha_i (1 - \epsilon_i) - \sum_i r_i \epsilon_i$

$$\mathcal{L}(w, b, \epsilon, a, r) = \frac{1}{2}\|w\|_2^2 + C\sum_i \epsilon_i - \|w\|_2^2 + \sum_i \alpha_i(1-\epsilon_i) - \sum_i r_i\epsilon_i$$

$$\mathcal{L}(w, b, \epsilon, a, r) = -\frac{1}{2}\|w\|_2^2 + C\sum_i \epsilon_i + \sum_i \alpha_i - \sum_i \alpha_i\epsilon_i - \sum_i r_i\epsilon_i$$

$$\mathcal{L}(w, b, \epsilon, a, r) = -\frac{1}{2}\|w\|_2^2 + C\sum_i \epsilon_i + \sum_i \alpha_i - \sum_i(\alpha_i + r_i)\epsilon_i$$

$$\begin{aligned}\mathcal{L}(w, b, \epsilon, a, r) &= -\frac{1}{2}\|w\|_2^2 + \sum_i \alpha_i \\ &= -\frac{1}{2}\sum_{j=1}^{p}\lambda_j^2 + \sum_i \alpha_i \\ &= -\frac{1}{2}\sum_{j=1}^{p}\left(\sum_{i=1}^{n}\alpha_i y_i x_{ij}\right)^2 + \sum_i \alpha_i \\ &= -\frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_i\alpha_k y_i y_k x_{ij}x_{kj} + \sum_i \alpha_i\end{aligned}$$

$r_i \geq 0, C = \alpha + r_i$ so $\alpha < C$

$$\mathcal{L}(\alpha) = -\frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{n}\sum_{k=1}^{n}\alpha_i\alpha_k y_i y_k x_{ij}x_{kj} + \sum_i \alpha_i \qquad s.t \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_i \alpha y_i = 0 \end{cases}$$

3.2 Answer

complementary slackness shows that:

$$\alpha_i\left[(1-\epsilon_i) - y_i(w^T x_i + b)\right] = 0$$

If α is non-zero, then we have:

$$(1-\epsilon_i) - y_i(w^T x_i + b) = 0$$

$$y_i(w^T x_i + b) = 1 - \epsilon_i$$

Because $\epsilon \geq 0$, it is intuitively true that:

$$y_i(w^T x_i + b) \leq 1$$

3.3 Answer

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import normalize
data = pd.read_csv('messidor.csv').values
training_x,testing_x,training_y,testing_y =train_test_split(data[:,:-1
],data[:,-1:], test_size=0.4,random_state = 32)
```

3.4 Answer

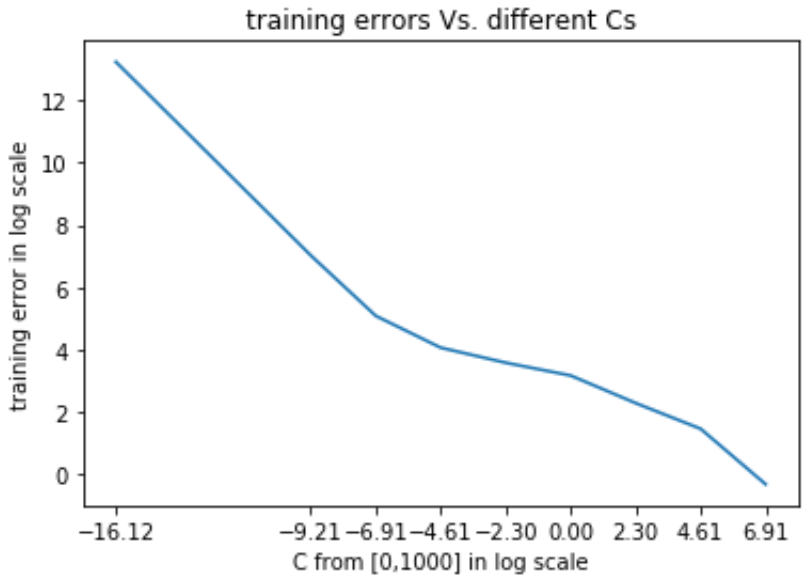
In [118]:
```
from sklearn.svm import SVC
sv = SVC(kernel = 'linear', C= 1e-20)
sv.fit(training_x,training_y.ravel())
train_accuracy = sv.score(training_x,training_y.ravel())
test_accuracy = sv.score(testing_x,testing_y.ravel())
weights = sv.coef_
print('training_accuracy is %0.3f' %(train_accuracy))
print('testing_accuracy is %0.3f' %(test_accuracy))
print('the weights are', weights)
```

```
training_accuracy is 0.523
testing_accuracy is 0.542
the weights are [[ 3.00000000e-20 -1.50000000e-19  4.06600000e-17  3
.44000000e-17
   2.86000000e-17  2.19900000e-17  1.69500000e-17  1.16500000e-17
   7.18046157e-18 -5.74552789e-18 -2.25135960e-19  1.84998684e-18
   1.79819602e-18  9.28043010e-19  4.34370770e-19  1.94914440e-19
   1.60247000e-21 -9.53390000e-22 -1.20000000e-19]]
```

3.5 Answer

In [127]:
```
Cs = [1e-7, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]
error = []
for C in Cs:
    sum = 0
    sv = SVC(kernel = 'linear', C=C)
    sv.fit(training_x,training_y.ravel())
    predictions = sv.predict(training_x)
    weights = sv.coef_
    i = 0
    for prediction in predictions:
        if training_y[i] != prediction:
            sum += prediction/np.linalg.norm(weights)
        i+=1
    error.append(sum)
```

In [143]:
```
logcs = np.log(Cs)
logerror = np.log(error)
plt.plot(logcs, logerror)
plt.xticks(logcs)
plt.xlabel('C from [0,1000] in log scale')
plt.ylabel('training error in log scale')
plt.title('training errors Vs. different Cs')
plt.show()
```

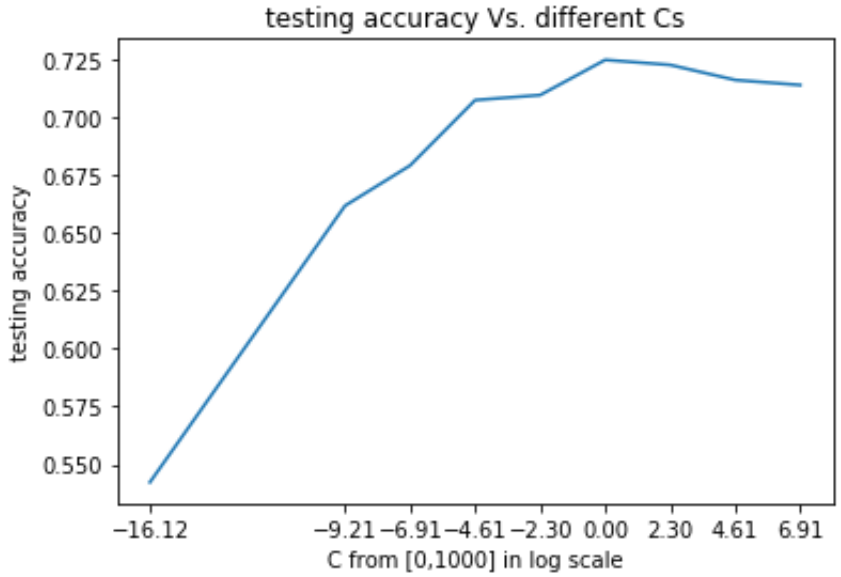


Observation: The training error becomes smaller when C gets larger. This is because for the non-seperable, if we do not add a large C, it will tend to seperate the points with big margins but there are mistakes. Therefore, once we add the large C, we get less error on the training set because we try to make it become the seperable case with smaller margin.

3.6 Answer

In [144]:
```
Cs = [1e-7, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]
accuracy = []
for C in Cs:
    sv = SVC(kernel = 'linear', C=C)
    sv.fit(training_x,training_y.ravel())
    test_accuracy= sv.score(testing_x,testing_y.ravel())
    accuracy.append(test_accuracy)
```

In [158]:
```
plt.plot(np.log(Cs), accuracy)
plt.xticks(logcs)
plt.xlabel('C from [0,1000] in log scale')
plt.ylabel('testing accuracy')
plt.title('testing accuracy Vs. different Cs')
plt.show()
print('the optimal C is', max(accuracy))
```



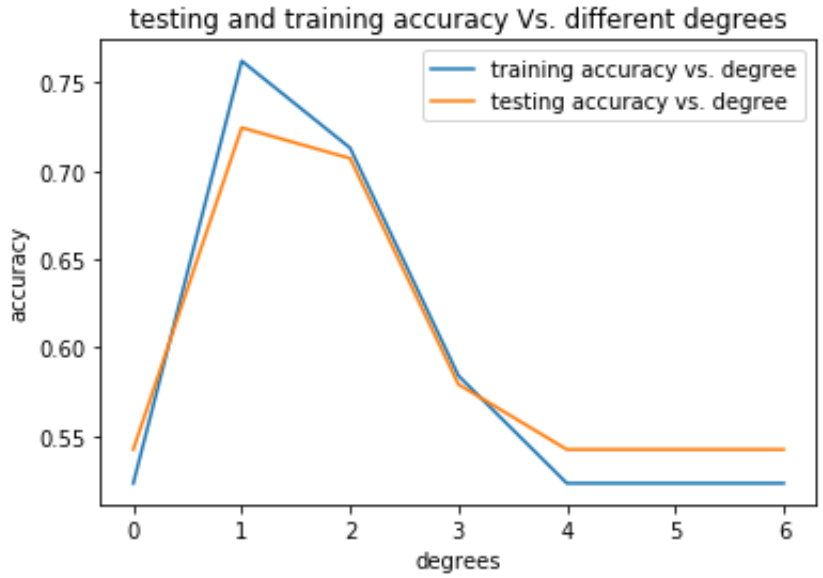

```
the optimal C is 0.7245119305856833
```

For the testing part, once we start to make c bigger, it will try to find a solution for this non-separable case. Therefore, the accuracy will increase under the growing of c. However, once the c increases the optimal value, the model start to overfit because it tries to fit everything in the training set by sacrificing the margin of the support vectors, therefore, we will have a little drop on the testing accuracy.

3.7 Answer

In [20]:
```
from sklearn.svm import SVC
from sklearn.preprocessing import normalize
## choose the optimal C =1
degree = [0, 1, 2, 3, 4, 5, 6]
test_accuracy = []
train_accuracy = []
for d in degree:
    sv = SVC(kernel = 'poly', degree=d, C=1, gamma='auto')
    sv.fit(normalize(training_x),training_y.ravel())
    te_accuracy= sv.score(normalize(testing_x),testing_y.ravel())
    tr_accuracy = sv.score(normalize(training_x),training_y.ravel())
    test_accuracy.append(te_accuracy)
    train_accuracy.append(tr_accuracy)
```

In [19]:
```
import matplotlib.pyplot as plt
plt.plot(degree, train_accuracy, label='training accuracy vs. degree')
plt.plot(degree,test_accuracy,label='testing accuracy vs. degree')
plt.xticks(degree)
plt.xlabel('degrees')
plt.ylabel('accuracy')
plt.title('testing and training accuracy Vs. different degrees')
plt.legend()
plt.show()
```



4. Kernels

4.1 Answer

According to the representer theorm: $f* = \sum_i \alpha_i K$

4.2 Answer

a. $k(x, z) = \alpha k_1(x, z) + \beta k_2(x, z)$ for $\alpha, \beta \geq 0$

Becase k_1 and k_2 are valid kernels, therefore they both have their feature Φ_1 and Φ_2 , inner product $\langle \rangle_{H_{k_1}}$ and $\langle \rangle_{H_{k_2}}$

therefore, we can expand $\alpha k_1(x, z)$ into $\alpha k_1(x, z) = \left\langle \sqrt{\alpha}\ \Phi_1(x), \sqrt{\alpha}\ \Phi_1(z) \right\rangle_{H_{k_1}}$

expand $\beta k_2(x, z)$ into $\beta k_2(x, z) = \left\langle \sqrt{\beta}\ \Phi_2(x), \sqrt{\beta}\ \Phi_2(z) \right\rangle_{H_{k_2}}$

Therefore, we have

$$\begin{aligned} k(x, z) &= \alpha k_1(x, z) + \beta k_2(x, z) \\ &= \left\langle \sqrt{\alpha}\ \Phi_1(x), \sqrt{\alpha}\ \Phi_1(z) \right\rangle_{H_{k_1}} + \left\langle \sqrt{\beta}\ \Phi_2(x), \sqrt{\beta}\ \Phi_2(z) \right\rangle_{H_{k_2}} \\ &= \left\langle \left[\sqrt{\alpha}\Phi_1(x), \sqrt{\beta}\Phi_2(x)\right], \left[\sqrt{\alpha}\Phi_1(z), \sqrt{\beta}\Phi_2(z)\right] \right\rangle_{H_{new}} \end{aligned}$$

Since $k(x, z)$ can be written in the form of an inner product, so it is proved to be a valid kernel

b. $k(x, z) = k_1(x, z)k_2(x, z)$

Based on the Mercer's Theorem,

$k_1(x,z) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\phi_i(z)$
$k_2(x,z) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(z)$

$$\begin{aligned} k(x,z) &= k_1(x,z)k_2(x,z) \\ &= \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\phi_i(z) \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(z) \\ &= \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \lambda_i\lambda_j \left[\phi_i(x)\psi_j(x)\right]\left[\phi_i(z)\psi_j(z)\right] \\ &= \sum_{i,j}^{\infty} \lambda_i\lambda_j \left[\phi_i(x)\psi_j(x)\right]\left[\phi_i(z)\psi_j(z)\right] \end{aligned}$$

So, use the Mercer's Therorem backwards again, we have:

$k(x,z) = \sum_p \kappa_p \Omega_p(x)\Omega_p(z)$, where $\kappa = \lambda_i\lambda_j \Omega(x) = \phi_i(x)\psi_j(x) \quad \Omega(z) = \phi_i(z)\psi_j(z)$
Therefore, $k(x,z)$ is a valid kernel

c. $k(x,z) = f(x)f(z)$ for $f : \mathcal{X} \to \mathbb{R}$

$k(x,z) = f(x)f(z) = f(z)f(x) = k(z,x)$

So k is symmetric Given x_i and z_i, define vector $\mathbf{f} = (f(x_1), \ldots f(x_n)), \mathbf{f} = (f(z_1), \ldots f(z_n))$

$$\mathbf{K} = \mathbf{f}\mathbf{f}^T$$

for any vector v, we have:

$$v^T \mathbf{K} v = (v^T f)^2 \geq 0$$

so K is positive definite
Then we have a Gramian matrix, and each entry is a inner product of $f(x_n)f(z_n)$ so $k(x,z)$ is a valid kernel.

d. $k(x,z) = f(k_1(x,z))$ for f a polynomial with positive cofficients

By proving a and b, we have:

$$k(x,z) = \alpha k_1(x,z) + \beta k_2(x,z)$$
$$k(x,z) = k_1(x,z)k_2(x,z)$$

Since f is a polynomial with postive cofficients, each term would have the form either be $\alpha k_1(x,z)$, $\alpha k_1(x,z)k_2(x,z)$ or constant (positive as specified in the prompt), and thanks to the proof from a and b. Any linear combination for $\alpha k_1(x,z)$ and $\alpha k_1(x,z)k_2(x,z)$ and constants would be vaild kernel.

e. $k(x,z) = e^{-\alpha||x-z||_2^2}$

$$k(x,z) = e^{-\alpha||x-z||_2^2}$$
$$= e^{-\alpha(||x||_2^2+||z||_2^2-2x^Tz)} = e^{-\alpha||x||_2^2}e^{(-\alpha||z||_2^2)}e^{(2\alpha x^Tz)}$$

let $g(x) = e^{-\alpha||x||_2^2}$ and $g(z) = e^{(-\alpha||z||_2^2)}$, therefore: $g(x)g(z)$ is a valid kernel proved in c. x^Tz is an inner product, so it is a kernel. Then: $2\alpha x^Tz = \alpha x^Tz + \alpha x^Tz$ given by the proof 1, is also a valid kernel.
Because $e^x = lim_{i\to\infty}\left(1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}\right)$
we have: $e^{k_1(x,z)} = lim_{i\to\infty}\left(1 + k_1(x,z) + \frac{k_1(x,z)^2}{2!} + \cdots + \frac{k_1(x,z)^n}{n!}\right)$
combine proof a and b we have $e^{k_1(x,z)}$ is also a valid kernel
So we have: $e^{(2\alpha x^Tz)}$ is also a valid kernel In conclusion: $e^{-\alpha||x||_2^2}e^{(-\alpha||z||_2^2)}e^{(2\alpha x^Tz)}$ is just the multiplication of two kernels, so according to proof c. It is a valid kernel.

In [ ]: