难理解的几点

一、KKT条件

http://blog.csdn.net/james\_616/article/details/72869015

从公式推导和几何意义解释：**α与C的大小关系**决定**支持向量的位置**。

二、blog

代码及理论（类似课本，代码很详细）

<http://www.hankcs.com/ml/support-vector-machine.html>

含有自己观点

<http://blog.csdn.net/u014688145/article/details/52963915>

理解SVM的三层境界

<http://blog.csdn.net/v_july_v/article/details/7624837#t14>

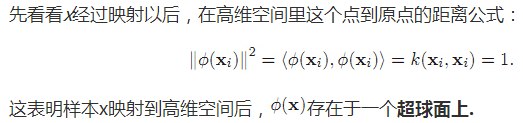
三、机器学习 径向基(Radial basis function)与RBF核函数 浅析

http://www.360doc.com/content/15/1214/00/28093736\_520273669.shtml

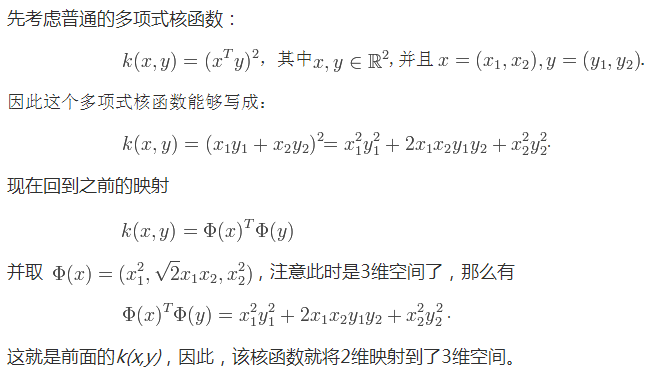
**核函数**：原本样本空间非线性问题，通过一种映射把它映射到高维空间使问题变得线性。用核函数代替内积，往往把高斯径向基函数作为核函数。



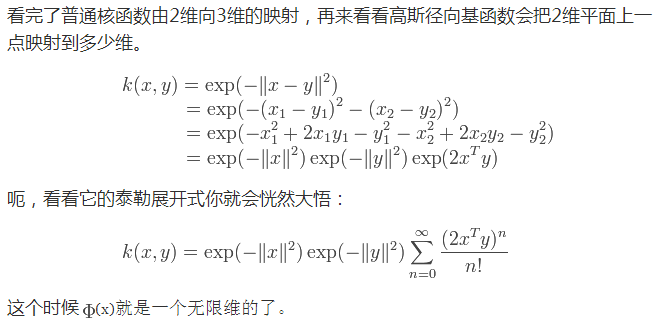
几何含义：



多项式核函数为例：



在原本样本空间，定义一种的运算→该运算结果为：高维样本空间的内积



四、代码解析

1.算法何时结束？停机条件是什么？

iter = 1，是无法使得m个样本点都满足KKT条件的；但随着迭代次数增多（alpha对的随机选择），可以使得m个样本点都满足KKT条件

从算法的角度来看，KKT条件便是求解目标函数极小值的充分必要条件。

2.第一个变量选择时，为什么首先遍历非边界样本？

因为随着多次子优化过程，边界变量倾向于留在边界，而非边界变量倾向于波动，这一步启发式的选择算法是基于节省时间考虑的，并且算法会一直在非边界变量集合上遍历，直到所有非边界变量都满足KKT条件（self-consistent）。

3.代码注意numpy的array与matrix区别

4.numpy中的nonzero

nonzeros(a)返回数组a中值不为零的元素的下标，它的返回值是一个长度为a.ndim(数组a的轴数)的元组，元组的每个元素都是一个整数数组，其值为非零元素的下标在对应轴上的值。

例如对于一维布尔数组b1，nonzero(b1)所得到的是一个长度为1的元组，它表示b1[0]和b1[2]的值不为0(False)。

对于二维数组b2，nonzero(b2)所得到的是一个长度为2的元组。它的第0个元素是数组a中值不为0的元素的第0轴的下标，第1个元素则是第1轴的下标，因此从下面的结果可知b2[0,0]、b[0,2]和b2[1,0]的值不为0：

>>> b2 = np.array([[True, False, True], [True, False, False]])

>>> np.nonzero(b2)

(array([0, 0, 1]), array([0, 2, 0]))

5.注意点

oS.alphas.A > 0 \* oS.alphas.A < oS.C

√ (oS.alphas.A > 0) \* (oS.alphas.A < oS.C)

6.画图代码模板

fig = plt.figure()

ax = fig.add\_subplot(111)

ax.scatter(xcord1,ycord1,marker='o',s=50,c='red')

plt.title('Support Vector Circled')

circle = Circle((sv[0,0],sv[0,1]))#

ax.add\_patch(circle)

ax.plot(x,y)

ax.axis([-2,12,-8,6])

plt.show()

**def calcEk(oS,k): #计算Ek**

fXk =

Ek =

return Ek

**def updateEk(oS,k):**

Ek = calcEk(oS,k)

ecache = [1,Ek]

**def selectJ(I,Ei,oS): #挑选使abs(Ei – Ej)波动最大的j**

maxK = -1;maxDeltaE = 0;Ej = 0

ecache[i] = [1,Ei]

有效位为1的误差列表validEcacheList

if len >1 :

对列表中每一个k，

排除k == i的情况

计算Ek、DeltaE比较

else:

随机选择j = **selectJrand**(i, oS.m);Ej = calcEkK(oS, j)

return maxK,Ej

**内层循环def innerLK(i, oS)：**

#计算Ei

if i不满足KKT条件：

（选择第二个alpha值）

j,Ej = **selectJ(I,Ei,oS)**

讨论yi和yj的值，确定L和H if L == H, return 0

eta =表达式if eta < 0, return 0

alphas[j] = 表达式

alphas[j] = **clipAlpha(oS.alphas[j],H,L)** #限定在盒子里

updateEk(oS,j)

if alphas[j]的变化<0.00001:return 0

更新alphas[i] = 表达式

updateEk(oS,i)

b1 = 表达式; b2 = 表达式;

考虑αi的取值，确定oS.b的值

return 1

else:return 0

**alphaPairs未变化(return 0)的条件：**

i满足KKT条件；

L == H; eta<0

alphas[j]取值波动太小，此时只能重新选择i

**def SMO(dataMat,labelMat,C,toler,maxIter):**

iter = 0;alphaPairChanged = 0;entireSet = True

**#停机条件：**

iter达到最大迭代次数;

即使没有达到maxIter,但遍历了间隔边界的Set,alphaPairs不变化

#第一轮entireSet = True遍历所有Set，找到一个不满足KKT条件的alpha

#第二轮翻转entireSet，只遍历间隔边界上nonBoundIs的支持向量点

如果第二轮的alphaPairChanged变化了，说明找到了alpha，继续下一轮

otherwise：翻转entireSet，即遍历所有Set寻找alpha

return b,alphas

def calcWs(alphas,dataMat, labelMat):

w = 表达式

主函数：

取α>0的样本点为支持向量，步骤如下：

svInd = α＞0的索引nonzero(alphas>0)[0]

dataSV = 数据集

labelSV =

print there are xx Support Vectors

predict误差

if abs(alphas[i]) > 0.0000001:

放到svList

from matplotlib.patches import Circle

def plotSVM(dataMat, labelMat, w, b, svList):

for i in range():

xPt = dataMat[:,0]

yPt = xx

if label == -1

xcard0 ycard0

else:

xcard1 ycard1

fig = plt.figure()

ax = fig.add\_subplot(111)

ax.scatter(x,y,marker='s', s=90,c=’red’)

ax.title(‘Support Vectors Circled’)

对svList中的每一个alpha: #圈出支持向量

circle = Circle((dataMat.loc[sv,0], dataMat.loc[sv,1]), 0.5, facecolor='none', edgecolor=(0,0.8,0.8), linewidth=3, alpha=0.5)

ax.add\_patch(circle)

w0=w[0]; w1=w[1]

x = np.arange(-2.0, 12.0, 0.1)

y = xxx #w0x1 + w1x2 + b = 0

ax.plot(x,y) #画出分界线

plt.show()

def kernelTrans(X, A, kTup):

"""

X：维度为m\*n的全部向量

A：某向量 / 均值

kTup：kTup[0]为类型 / kTup[1]为方差

"""

if kTup[0]=='lin': #线性核函数

K = X \* A.T

elif kTup[0]=='rbf': #高斯径向基核函数

对每个样本求top

K = exp(-\*\*)

else:

raise NameError(‘We Have a Problem -- \

That Kernel is not recognized’)

return K

注意，求解predict可以只考虑支持向量，因为alpha＞0，其余的相乘时都消掉了。alpha＞0称为支持向量，因为只有大于0的才决定决策边界！