机器学习 第10章 作业

10.1

```
import numpy as np
import matplotlib.pyplot as plt

#设置绘图时显示中文
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False

def kNN(X,Y,Xpre,k,p=2):
    # k近邻算法
    # X：样本数据
    # Y：样本标记
    # Xpre：待预测样本
    # k：k近邻的k值
    # p:计算距离所采用的闵可夫斯基距离的p值
    mt,n=X.shape              #训练样本数和特征数
    Xpre=np.array(Xpre).reshape(-1,n)
    mp=Xpre.shape[0]          #预测样本数
    dist=np.zeros([mp,mt])    #存储预测样本和训练样本之间的距离
    for i in range(mt):
        dist[:,i]=(((abs(Xpre-X[i]))**p).sum(axis=1))**(1/p)
    neighbor=np.argsort(dist,axis=1)    #训练样本按距离远近排序的索引号
    neighbor=neighbor[:,:k]             #只取前k个作为最近邻
    Ypre=Y[neighbor]
    return (Ypre.sum(axis=1)>=0)*2-1    #西瓜3.0α仅两类，故可如此计算

# 西瓜3.0α 样本数据
X=np.array([[0.697,0.46],[0.774,0.376],[0.634,0.264],[0.608,0.318],[0.556,0.215],
   [0.403,0.237],[0.481,0.149],[0.437,0.211],[0.666,0.091],[0.243,0.267],
   [0.245,0.057],[0.343,0.099],[0.639,0.161],[0.657,0.198],[0.36,0.37],
   [0.593,0.042],[0.719,0.103]])
Y=np.array([1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1])

# 执行kNN算法
# 尝试 k=1,3,5,p=1,2,30的不同情况
ks=[1,3,5]
ps=[1,2,50]  #p=1为曼哈顿距离，p=2为欧式距离，p=50(→∞)为切比雪夫距离
for i,k in enumerate(ks):
    for j,p in enumerate(ps):
        # kNN算法预测结果
        x0=np.linspace(min(X[:,0]),max(X[:,0]),60)
        x1=np.linspace(min(X[:,1]),max(X[:,1]),60)
        X0,X1=np.meshgrid(x0,x1)
        Xpre=np.c_[X0.reshape(-1,1),X1.reshape(-1,1)]
        Ypre=kNN(X,Y,Xpre,k,p).reshape(X0.shape)
        # 画图
        plt.subplot(len(ks),len(ps),i*len(ps)+j+1)
        #plt.axis('equal')
        plt.title('k=%d,p=%d'%(k,p))
        plt.xlabel('密度')
```

```
        plt.ylabel('含糖率')
        # 画样本点
        plt.scatter(X[Y==1,0],X[Y==1,1],marker='+',s=30,label='好瓜')
        plt.scatter(X[Y==-1,0],X[Y==-1,1],marker='_',s=30,label='坏瓜')
        # 画决策树边界 (直接根据教材上图4.10和4.11确定边界曲线坐标)
        plt.plot([0.381,0.381,0.56,0.56,max(X[:,0])],
                 [max(X[:,1]),0.126,0.126,0.205,0.205],'k',label='决策树边界')
        # 画kNN边界
        plt.contour(X0,X1,Ypre,1,colors='r',s=2)
plt.show()
```

讨论：

1. 当k=1时训练误差为零；当k越来越大时，分类边界趋向平缓，会出现训练样本误分类的情况，此时为欠拟合。可见，k越小越容易过拟合，k越大越容易欠拟合。
2. 不同p值的距离计算下，差别不太明显。

10.2

证明：首先来理解一下这两个期望错误率，它们都表示成"错误率=1-正确率"的形式，因此问题转化成理解正确率。对于预测样本x，贝叶斯最优分类器的决策结果c^*，如果预测正确，意味着这个样本的真实类别刚好也是c^*，而该样本以$P(c^*|x)$的概率属于c^*类别，因此预测正确的概率即为$P(c^*|x)$，而错误率$err^* = 1 - P(c^*|x)$。
最近邻分类器的决策结果等于近邻样本z相同的类别，假设这个共同的类别为c，如果这个决策结果是正确的，意味着x样本和z样本刚好都属于c类，这个事件发生的概率是$P(c|x)P(c|z)$。考虑各种不同的c值，把它们加起来便是总的期望正确率$=\sum_c P(c|x)P(c|z)$，因此错误率
$err = 1 - \sum_c P(c|x)P(c|z)$。

接下来证明上面的不等式：

$$
\begin{aligned}
\text{err} &= 1 - \sum_c P(c|x)P(c|z) \\
&\geq 1 - \sum_c P(c^*|x)P(c|z) \\
&= 1 - P(c^*|x) \cdot \sum_c P(c|z) \\
&= 1 - P(c^*|x) \\
&= err*
\end{aligned}
$$

第2行利用了关系$P(c|x) \leq P(c^*|x)$,第4行利用了关系$\sum_c P(c|z) = 1$。
不等式左半边得证。

$$
\begin{aligned}
err &= 1 - \sum_c P(c|x)P(c|z) \\
&\approx 1 - \sum_c P^2(c^{|x)} \\
&= 1 - P^2(c^*|x) - \sum_{c \neq c^*} P^2(c|x) \\
&\leq 1 - P^2(c^*|x) - \frac{[\sum_{c \neq c^*} P(c|x)]^2}{|y| - 1} \\
&= 1 - (1 - err^*)^2 - \frac{(err^*)^2}{|y| - 1} \\
&= err^*(2 - \frac{|y|}{|y| - 1} err^*)
\end{aligned}
$$

第4行利用了不等式关系：$\frac{\sum a_i^2}{n} \geq (\frac{\sum a_i}{n})^2, a_i \geq 0, i = 1, 2, \cdots, n$; 第5行利用了关系

$\sum_c \neq c^* P(c|x) = 1 - P(c^*|x) = err^*.$
不等式右半边得证。

10.3

答：相当于将X变为X'：

$$
\begin{aligned}
X' &= XH \\
&= X(I - \frac{1}{m}11^T) \\
&= X - \frac{1}{m}X11^T \\
&= X - \frac{1}{m}[x_1, \quad x_2, \quad \cdots] \begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix} [1 \quad 1 \quad \cdots] \\
&= X - \frac{1}{m}\sum_i x_i [1 \quad 1 \quad \cdots] \\
&= X - \bar{x}[1 \quad 1 \quad \cdots] \\
&= [x_1 - \bar{x}, \quad x_2 - \bar{x}, \quad \cdots]
\end{aligned}
$$

其效果便是中心化$x'_i = x_i - \bar{x}$。

10.4

答：首先看一下，确实可以通过SVD(奇异值)分解得到协方差矩阵的特征值和特征向量。由附录A.33式有，任意实矩阵$A \in R^{m \times n}$都可以分解为$A = U\Sigma V^T$。

其中U和V分别为m阶和n阶的酉矩阵，Σ是m*n矩阵，对角元以外元素均为零，于是有：

$$AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$$
$$A^T A = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T$$

因此，U的列向量是AA^T的特征向量，V的列向量是$A^T A$的特征向量，Σ矩阵的非零对角元σ_{ii}的 平 方 即 为 AA^T和$A^T A$的共同非零待征值。

在前面原理介绍部分采用的10.2和10.4等图中，数据维度d=2或3，而样本数m远大于数据维数，m>>d。然而在实际情况中，既然需要降维，通常维度d很大，比如对于100张100*100的图片，m=100，d=10000，此时的协方差矩阵XX^T的shape为$R^{10000 \times 10000}$，矩阵维度较大。而$X \in R^{10000 \times 100}$，对其进行SVD(奇异值)分解的计算成本较低一些。