DSCI6001P 数据科学基础

作业3 集成、聚类、贝叶斯

Problem 1

1. K-mediods 算法描述：

a) 首先随机选取一组聚类样本作为中心点集

b) 每个中心点对应一个簇

c) 计算各样本点到各个中心点的距离（如欧几里德距离），将样本点放入距离中心点最短的那个簇中

d) 计算各簇中，距簇内各样本点距离的绝度误差最小的点，作为新的中心点

e) 如果新的中心点集与原中心点集相同，算法终止；如果新的中心点集与原中心点集不完全相同，返回 b)

问题：

a) 阐述 $K-mediods$算法和$K-means$ 算法相同的缺陷

b) 阐述 $K-mediods$ 算法相比于$K-means$ 算法的优势

c) 阐述 $K-mediods$ 算法相比于$K-means$ 算法的不足

d) 思考一个自动确定聚类个数的改进 $K-means$ 算法，或者说如何确定 kmeans 聚类个数（伪代码或者算法描述）

答：

问题（a）：

分析可知，两者共同的缺点有参数k需要调整，算法过程无法确定k的最佳。同时，初始化中心点的选择过于随机，可能选择很差的中心点只能达到局部最优解（比如，$K-means++$进行了这个方面的优化）。

问题（b）：

$K-mediods$中的选择的中心点集必须是聚类的样本点中的值，而$K-means$直接对所有样本点取均值来进行选择（可以不是样本点中的值）。$K-mediods$以其他点到特定点的距离之和作为优化函数可以在一定程度上削弱异常数据点对于聚类结果的影响。而$K-means$的缺点是鲁棒性不足，对于噪声和离群点比较敏感。

问题（c）：

$K-mediods$ 算法相比于$K-means$ 算法的不足，$K-mediods$中的选择的中心点集必须是聚类的样本点中的值，而$K-means$直接对所有样本点取均值来进行选择（可以不是样本点中的值）。因此，可知$K-mediods$算法的运算速度会更慢一

些，因为它判断新的中心点时，需要找出簇内到各个样本点距离的绝对误差最小的点，这个过程的时间复杂度为$O(n^2)$，而$K-means$ 直接计算样本的均值作为中心点，这个过程时间复杂度为$O(n)$。其中n为样本点的数目。

问题（d）：

思考发现，最佳的k可以通过聚类质量来得到，而聚类的质量可以通过聚类无监督评价指标例如$silhouette$系数来得到。基于这种考虑，可以设计出算法找到最佳的$silhouette$系数来得到最佳的k。

```
Input: Data, left, right #假设在left到right范围内选取最优的k
Ouput: Cluster, k
for i in range(left, right + 1):
    do K-means(i) get Cluster
    calulate_silhouette(Cluster)
    if silhouette < 之前计算出的所有silhoutte:
        更新k = i, Cluster
return k, Cluster
```

除此之外，实验中可以发现，$K-means$ 的簇内误差平方和随着k的增加会出现一个拐点，因此可以在算法执行过程之中，对簇内误差平方和与上次的差值进行判断来确定k。

```
Input: Data, difference
Ouput: Cluster, k
i = 2
while i:
    do K-means(i) get Cluster
    calulate_deviation(Cluster)
    if deviation_before - deviation_now < difference:
        更新k = i, Cluster
        break
return k, Cluster
```

Problem 2

2. 集成学习：

(1) 试析随机森林为何比决策树 Bagging 集成的训练速度更快？

(2) 集成学习中多样性增强的方法有哪些？分别阐述这些方法适用的前提。

(3) Bagging 能否提升朴素贝叶斯分类的性能？为什么？

(4) 分析 GradientBoosting [Friedman, 2001]和 AdaBoost 的异同？

答：

问题（1）：

随机森林在决策树的训练过程之中加入了随机属性选择的策略，这个策略大大减少了寻找最佳划分属性所需要的计算量。而决策树的Bagging集成没有对该过程进行优化，只是随机采样并学习弱分类器来集成，在对于最佳划分属性的选择是非常耗时间的。

问题（2）：

方法1：选择不同的样本。例如使用$Bootstrap$方法，该方法可以通过选择不同的样本得到不同的学习器。该方法使用的前提是学习算法是对于样本扰动的敏感的算法，例如决策树的算法。而对于SVM,，KNN这类不适用。
方法2：参数扰动。对于算法的参数进行更改，比如神经网络中的权值和神经元数目等等。通过这种方法，可以产生差别较大的学习器。但是这种方法适用于对于参数扰动敏感的模型。
方法3：随机选择输入属性。可以使用随机子空间法从有大量冗余属性的数据集中选择部分的属性为子集来训练学习器。这个方法对于属性较少的数据集并不适用。
方法4：对输出表示进行扰动。可以对训练集的标签进行更改。例如，使用翻转法或者输出调制法。这个方法适用于对于输出表示扰动敏感的算法。这样可以达到多样性。

问题（3）：

不能。$bagging$可以通过融合多个容易过拟合的模型比如决策树和神经网络，来降低方差。而朴素贝叶斯模型本身的拟合能力不足，方差本就不大，同时它假设样本各个特征相互独立，其主要的误差来自于偏差。因此使用$bagging$方法提升的效果也并不明显。$Boosting$方法主要偏重于降低偏差，因此他对于朴素贝叶斯模型应该会有效果。

问题（4）：

相同之处：两者都是将弱模型进行集成得到一个更强的模型。不同之处：$GradientBoosting$的训练过程可以看作对可导的目标函数的优化过程。它主要是通过梯度来发现模型的不足来进行相关优化。而$AdaBoost$是通过提高错分的数据点的权重从而定位模型的不足进行相关优化。因此两者的优化目标存在差别，$GradientBoosting$优化目标更加多样。

Problem 3

3. 假设数据挖掘的任务是将如下的 8 个点(用(x，y)代表位置)聚类为 3 个簇。
$A_1(2,10), A_1(2,5), A_3(8,4), B_1(5,8), B_2(7,5), B_3(6,4), C_1(1,2), C_2(4,9)$
距离函数是欧氏距离。假设初始我们选择A_1，B_1和C_1分别为每个簇的中心，用 K-均值算法给出:
(a)在第一轮执行后的 3 个簇中心。
(b)最后的 3 个簇。
答：

问题（a）：

由公式计算可知，第一轮执行结束的三个中心坐标为$(2,10),(6,6),(1.5,3.5)$

问题（b）：最后的3个簇为：

$cluster0: A_1(2,10), B_1(5,8), C_2(4,9)$

$cluster1: A_3(8,4), B_2(7,5), B_3(6,4)$

$cluster2: A_1(2,5), C_1(1,2)$

```python
import numpy as np
point = np.array([[2, 10], [2, 5], [8, 4], [5, 8], [7, 5], [6, 4], [1, 2], [4, 9]])
core = np.array([[2.0, 10.0], [5.0, 8.0], [1.0, 2.0]])
label = np.array([-1] * 8)
label[0] = 0
label[3] = 1
label[6] = 2
newlabel = np.array([0] * 8)
iteration = 1

while((label != newlabel).any()):
    print(f'iteration:{iteration}')
    iteration += 1
    label = newlabel.copy()
    for i in range(len(point)):
        tmp = 0x3f3f3f3f
        id = 0
        for j in range(len(core)):
            dis = np.sqrt(np.sum((point[i] - core[j]) ** 2))
            if(tmp > dis):
                tmp = dis
                id = j
        newlabel[i] = id
    summ = np.array([[0, 0],[0, 0],[0, 0]])
    count = np.array([0.0] * 3)
    for i in range(len(point)):
        for j in range(len(core)):
            if newlabel[i] == j:
                summ[j] += point[i]
                count[j] += 1
                break
    for j in range(len(core)):
        if(count[j] == 0):
            print(count[j])
        core[j] = summ[j] / float(count[j])
    print(f'core: {core}')
for i in range(8):
    print(f'point:{point[i][0]}, {point[i][1]}. cluster id: {label[i]}')
```

```
#output共迭代了4轮
iteration:1
core: [[2.  10.] [6.   6.] [1.5  3.5]]
iteration:2
core: [[3.   9.5 ] [6.5  5.25] [1.5  3.5]]
iteration:3
core: [[3.66666667 9.] [7. 4.33333333] [1.5 3.5]]
iteration:4
core: [[3.66666667 9.] [7. 4.33333333] [1.5 3.5]]
point:2, 10. cluster id: 0
point:2, 5. cluster id: 2
point:8, 4. cluster id: 1
point:5, 8. cluster id: 0
point:7, 5. cluster id: 1
point:6, 4. cluster id: 1
point:1, 2. cluster id: 2
point:4, 9. cluster id: 0
```

Problem 4

4. 假设你打算在一个给定的区域分配一些自动取款机(ATM)，使得满足大量约束条件。住宅或工作场所可以被聚类以便每个簇被分配一个 ATM。然而. 该聚类可能被两个因素所约束：(1)障碍物对象, 即有一些可能影响 ATM 可达性的桥梁、河流和公路。(2)用户指
定的其他约束，如每个 ATM 应该能为10000户家庭服务。在这两个约束条件下，怎样修改聚类算法(如 K-均值)来实现高质量的聚类？
答：

可以考虑找到许多内部无障碍的类簇进行预处理。在有障碍物对象的条件下，我们需要对距离及逆行重新定义（例如，两个类之间通过桥梁相连接，这两个类之间的点的距离是点到桥的距离再加上桥到点的距离）。利用这个可达的距离来替代传统的欧式距离，这样可以更好地刻画出场所与场所之间的连通性。同时，这里存在最大容量的约束。对于K-均值算法，如果在迭代过程之中，存在容量大于10000的类簇。我们可以考虑将离中心点最近的10000个场所保留，然后将多余的部分基于距离分给其它容量小于10000的类簇之中，循环迭代，直至所有类簇的大小都小于10000，然后再进行下一轮。每个ATM机放在聚类的中心处。

```
#在每轮K-means迭代的最后加上这个代码
process(cluster):
    while(1):
        for c in cluster:
            ok = 0
            if |c| > 10000:
                ok = 1
                calculate the top-10000 closest point as set S
                p = c - s
                reallocate points in p (put them in the closest cluster)
            if(ok == 0):
                stop
```

Problem 5

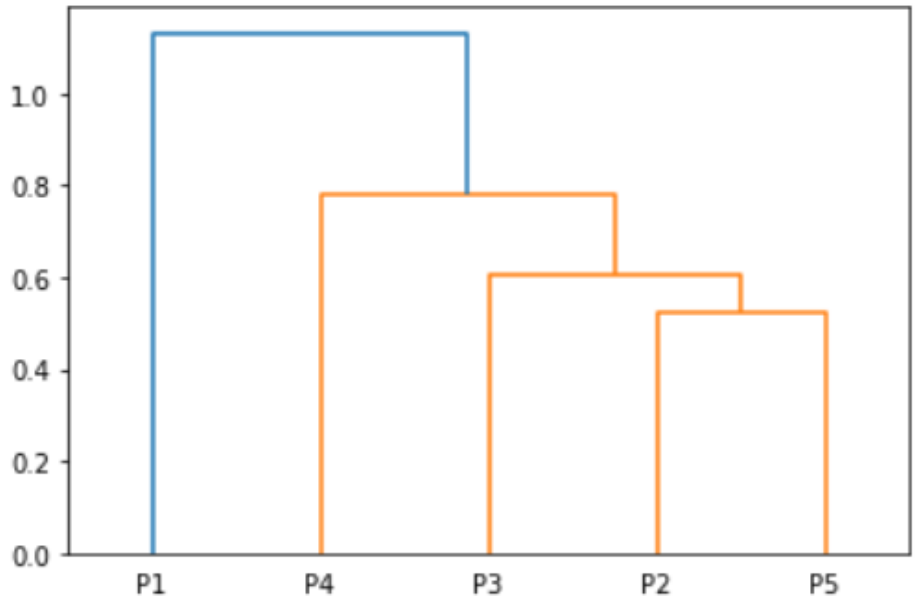
5. 使用如下表中的相似度矩阵进行单链和全链层次聚类。绘制树状图显示结果，树状图应清楚地显示合并的次序。

	P1	P2	P3	P4	P5
P1	1.00	0.10	0.41	0.25	0.35
P2	0.10	1.00	0.64	0.47	0.98
P3	0.41	0.64	1.00	0.44	0.85
P4	0.25	0.47	0.44	1.00	0.76
P5	0.35	0.98	0.85	0.76	1.00

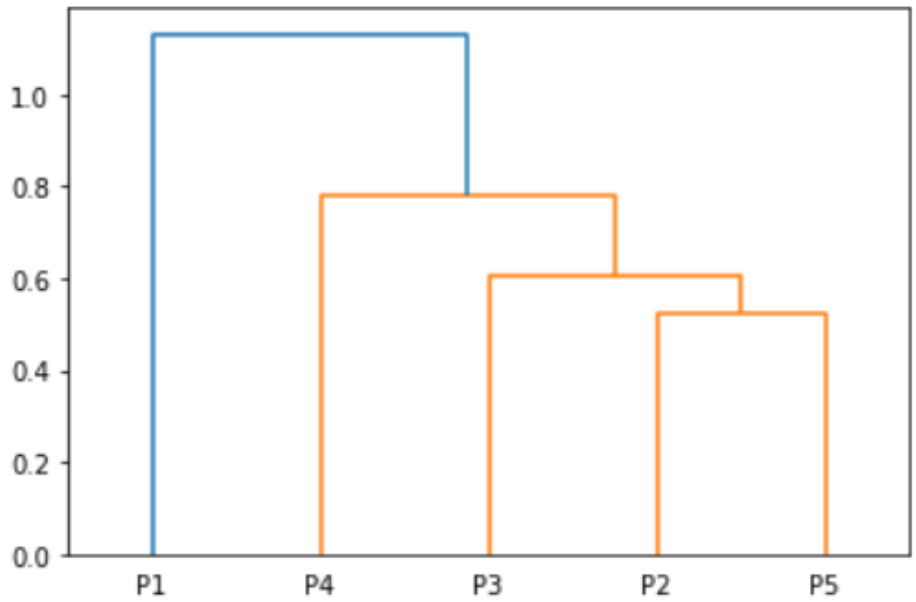
答：

（1）单链。定义临近度为不同两个簇最近的点的距离。首先，根据表发现P_2和P_5

的相似性为0.98最大，应该首先合并得到(P_2, P_5)，更新(P_2, P_5)与P_1，P_3和P_4的相似性分别为0.1，0.64，0.47。接着找(P_2, P_5)，P_1，P_3和P_4中两两之间相似度最大的进行合并。通过计算可知，(P_2, P_5)与P_3最近相似性为$sim((2,5),3) = 0.64$，将(P_2, P_5)与P_3进行合并，并更新(P_2, P_5, P_3)与P_1和P_4的相似性分别为0.1和0.44。同理，找(P_2, P_5, P_3)，P_1和P_4中最近的类进行合并发现(P_2, P_5, P_3)与P_4的相似性最大为$sim((2,5,3),4) = 0.44$。最后将(P_2, P_5, P_3, P_4)与P_1进行合并得到最终的层次聚类结果。



（2）**全链层次**。定义临近度为不同两个簇最远的点的距离。首先，根据表发现P_2和P_5的相似性为0.98最大，应该首先合并得到(P_2, P_5)，更新(P_2, P_5)与P_1，P_3和P_4的相似性分别为0.35，0.85，0.76。。接着找(P_2, P_5)，P_1，P_3和P_4中最近的两个类进行合并。通过计算可知，(P_2, P_5)与P_3最近相似性为$sim((2,5),3) = 0.85$，并更新(P_2, P_5, P_3)与P_1和P_4的相似性分别为0.41和0.76。同理，找(P_2, P_5, P_3)，P_1和P_4中最近的类进行合并发现(P_2, P_5, P_3)与P_4的相似性最大为$sim((2,5,3),4) = 0.76$。最后将(P_2, P_5, P_3, P_4)与P_1进行合并得到最终的层次聚类结果。很容易发现，使用单链和全链层次聚类得到的聚类结果是一样的。



```python
import numpy as np
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt
labels = ['P1', 'P2','P3','P4','P5']
test = np.array(
[[1.00,   0.10,    0.41,  0.25,   0.35,   0.35],
 [0.10,   1.00,    0.64,  0.47,   0.47,   0.98],
 [0.41,   0.64,    1.00,  0.44,   0.44,   0.85],
 [0.25,   0.47,    0.44,  1.00,   1.00,   0.76],
 [0.35,   0.98,    0.85,  0.76,   0.76,   1.00]]
)
def hierarchy(test, method):
    m = linkage(test, method=method)
    dendrogram(m, labels=labels, leaf_font_size=10)
    plt.show()
hierarchy(test, method='single')
hierarchy(test, method='complete')
```

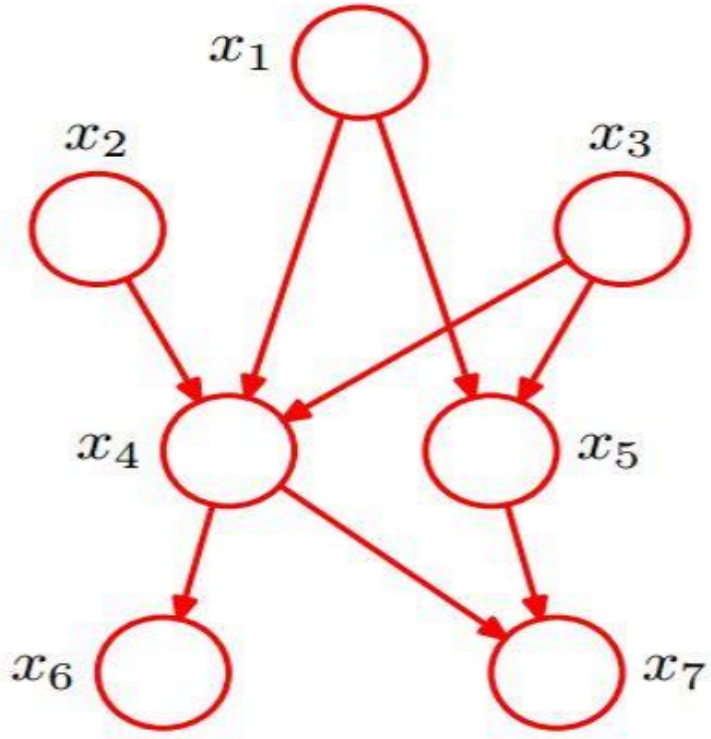
Problem 6

6. 给定一个贝叶斯网络如下图所示：

（1） 在给定x_1, x_3的情况下，x_5, x_6是条件独立的吗？

（2） 在给定x_2, x_3的情况下，x_5, x_6是条件独立的吗？

（3） 写出$x_1, x_2, \ldots x_7$的联合概率分布



答：

问题（1）： 在给定x_1, x_3的情况下，x_5，x_6是条件独立的。首先将，上述的有向图贝叶斯网络可以转化为$MoralGraph$形式，即在x_1和x_3，x_1和x_2以及x_4和x_5之间添加一条边并把有向图转化为无向图。观察$MoralGraph$很容易发现：

$x_5 \perp x_6 | (x_1, x_3, x_4)$，因为$x_5$和$x_6$被$x_1, x_3, x_4$分隔开。同理$x_4 \perp x_5 | (x_1, x_3)$，这样的话很容易发现：$P(x_5, x_6 | x_1, x_3) = \frac{P(x_5, x_6 | x_1, x_3, x_4) \times P(x_4 | x_1, x_3)}{P(x_4 | x_1, x_3, x_5, x_6)} =$

$\frac{P(x_5|x_1,x_3,x_4)\times P(x_6|x_1,x_3,x_4)\times P(x_4|x_1,x_3)}{P(x_4|x_1,x_3,x_5,x_6)} = \frac{P(x_5|x_1,x_3)\times P(x_6|x_1,x_3)\times P(x_4|x_1,x_3,x_6)}{P(x_4|x_1,x_3,x_5,x_6)} =$
$P(x_5|x_1,x_3)\times P(x_6|x_1,x_3)$

综上，在给定x_1,x_3的情况下，x_5，x_6是条件独立的。

问题（2）：在给定x_2,x_3的情况下，x_5和x_6不是条件独立的。因为没有给定x_1使得x_4和x_5不是条件独立的，从而造成x_6不是条件独立的。

问题（3）：$x_1,x_2,\ldots x_7$的联合概率分布为：

$P(x_1)P(x_2)P(x_3)P(x_4|x_1,x_2,x_3)P(x_5|x_1,x_3)P(x_6|x_4)P(x_7|x_4,x_5)$

Problem 7

7. 试由下表的训练数据学习一个朴素贝叶斯分类器并确定 x = （2，S，T）的类判别结果y。表中 X（1），X（2） X（3）为特征，Y 为类标记。

	1	2	3	4	5	6	7	8	9	10
$X^{(1)}$	1	1	1	2	2	1	2	2	3	3
$X^{(2)}$	S	M	M	S	S	L	M	M	L	S
$X^{(3)}$	T	T	F	F	F	T	F	T	T	F
Y	-1	-1	1	1	-1	-1	-1	1	1	1

答：

$$P(Y=1)=\frac{5}{10},P(Y=-1)=\frac{5}{10} \tag{1}$$

$$P(X^{(1)}=1|Y=1)=\frac{1}{5},P(X^{(1)}=2|Y=1)=\frac{2}{5},P(X^{(1)}=3|Y=1)=\frac{2}{5} \tag{2}$$

$$P(X^{(1)}=1|Y=-1)=\frac{3}{5},P(X^{(1)}=2|Y=-1)=\frac{2}{5},P(X^{(1)}=3|Y=-1)=\frac{0}{5} \tag{3}$$

$$P(X^{(2)}=S|Y=1)=\frac{2}{5},P(X^{(2)}=M|Y=1)=\frac{2}{5},P(X^{(2)}=L|Y=1)=\frac{1}{5} \tag{4}$$

$$P(X^{(2)}=S|Y=-1)=\frac{2}{5},P(X^{(2)}=M|Y=-1)=\frac{2}{5},P(X^{(2)}=L|Y=-1)=\frac{1}{5} \tag{5}$$

$$P(X^{(3)}=T|Y=1)=\frac{2}{5},P(X^{(3)}=F|Y=1)=\frac{3}{5} \tag{6}$$

$$P(X^{(3)}=T|Y=-1)=\frac{3}{5},P(X^{(3)}=F|Y=-1)=\frac{2}{5} \tag{7}$$

$$P(Y=1)P(X^{(1)}=2|Y=1)P(X^{(2)}=S|Y=1)P(X^{(3)}=T|Y=1)=\frac{4}{125} \tag{8}$$

$$P(Y=-1)P(X^{(1)}=2|Y=-1)P(X^{(2)}=S|Y=-1)P(X^{(3)}=T|Y=-1)=\frac{6}{125} \tag{9}$$

上面的计算方法没有考虑拉普拉斯修正，考虑拉普拉斯修正的计算方法如下。

$$P(Y=1)=\frac{5+1}{10+2},P(Y=-1)=\frac{5+1}{10+2} \tag{10}$$

$$P(X^{(1)}=1|Y=1)=\frac{1+1}{5+3},P(X^{(1)}=2|Y=1)=\frac{2+1}{5+3},P(X^{(1)}=3|Y=1)=\frac{2+1}{5+3} \quad (11)$$

$$P(X^{(1)}=1|Y=-1)=\frac{3+1}{5+3},P(X^{(1)}=2|Y=-1)=\frac{2+1}{5+3},P(X^{(1)}=3|Y=-1)=\frac{0+1}{5+3} \quad (12)$$

$$P(X^{(2)}=S|Y=1)=\frac{2+1}{5+3},P(X^{(2)}=M|Y=1)=\frac{2+1}{5+3},P(X^{(2)}=L|Y=1)=\frac{1+1}{5+3} \quad (13)$$

$$P(X^{(2)}=S|Y=-1)=\frac{2+1}{5+3},P(X^{(2)}=M|Y=-1)=\frac{2+1}{5+3},P(X^{(2)}=L|Y=-1)=\frac{1+1}{5+3} \quad (14)$$

$$P(X^{(3)}=T|Y=1)=\frac{2+1}{5+2},P(X^{(3)}=F|Y=1)=\frac{3+1}{5+2} \quad (15)$$

$$P(X^{(3)}=T|Y=-1)=\frac{3+1}{5+2},P(X^{(3)}=F|Y=-1)=\frac{2+1}{5+2} \quad (16)$$

$$P(Y=1)P(X^{(1)}=2|Y=1)P(X^{(2)}=S|Y=1)P(X^{(3)}=T|Y=1)=\frac{27}{896} \quad (17)$$

$$P(Y=-1)P(X^{(1)}=2|Y=-1)P(X^{(2)}=S|Y=-1)P(X^{(3)}=T|Y=-1)=\frac{36}{896} \quad (18)$$

综上，$x=$（2，S，T）的类判别结果为-1