# K-近邻分类方法

- ✓ 简单概念
- ✓ K-近邻基本思路
- ✓ K-最近邻算法
- ✓ K-近邻分类方法也可作为预测方法
- ✓ 基于距离的分类方法

# 简单概念

K-近邻分类方法特点

1. 不是事先通过数据来学好分类模型，再对未知样本分类，而是存储带有标记的样本集，给一个没有标记的样本，用样本集中k个与之相近的样本对其进行即时分类。由于没有事先学习出模型，所以把它称作基于要求或懒惰的学习方法。
2. 这是一种基于示例的学习方法，一种基于类比的学习方法。
3. K-近邻就是找出k个相似的实例来建立目标函数逼近。这种方法为局部逼近，复杂度低。

# 简单概念--相似

对于距离的计算方法有许多：

设样本为 $X=(x_1,x_2,\ldots x_n)$

明考斯基距离： $d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+\ldots+|x_{i_p}-x_{j_p}|^q)}$

曼哈坦距离： $d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+\ldots+|x_{i_p}-x_{j_p}|$

欧氏距离： $d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+\ldots+|x_{i_p}-x_{j_p}|^2)}$

距离近就相似

# K-近邻基本思路

1. 存储一些标记好的样本集（样本都分好了类）
2. 一个未知类的样本（要对其分类）
3. 逐一取出样本集中的样本，与未知类样本比较，找到K-个与之相近的样本，就用这K-个样本的多数的类（或类分布）为未知样本定类。
4. 在样本集为连续值时，就用K-个样本的平均值为未知样本定值。

# K-最近邻算法

样本：用 n 维数值属性表示

每个样本为n维空间一个点

X=(x1, x2, ..... xn)

Y=(y1, y2, ..... yn)

度量：点之间的距离（关系）表示

$$d(X,Y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

# K-近邻算法

输入：

T　　//训练数据( 带有类标记的样本)

K　　//邻居的数目（给定k个近邻）

t　　//将要被分类的元组

输出：

c//元组t被分配的类别

算法://利用K-近邻（k-NN）算法对元组进行分类

```
//对于元组t发现的邻居集合
for each d∈T do
    if |N|≤K, then
        N=N∪{d};
    else
      if u∈N such that sim(t, u) ≤sim(t, d),
      then
            begin
              N=N-{u};//去掉与 t  距离大的u;
              N=N∪{d};//加进与 t 距离小的d;
            end
     //发现分类的类别
  c=class  to  which  the  most  u∈N  are
   classified;//N中的最多的类 c 赋给 t
```

从数据集T中不断取d
一直取出K个

将t与数据集中都比一遍，留
留下k个与之最小距离的元组

# K-近邻方法的优缺点

优点：

（1）易于编程，且不需要优化和训练

（2）当样本增大到一定容量，k也增大到合适的程度，k-近邻的误差可与贝叶斯方法相比。

缺点：

（1）在高维和数据质量较差时，k-近邻方法表现不好。

（2）当n个训练样本，n大时，计算时间太大。

如计算一个点要p次操作，每次查询都要np次计算，时间复杂度为O(np)。往往用户难以接受。

K-近邻方法对k的选择也是要靠经验，也取决于要处理的问题与背景。

# 基于距离的分类方法

- 近邻的含义？

  用什么方法来判断近邻也因问题不同而不同。

- 距离的计算?

  用什么方法来判断距离，距离怎样计算，这些都是因问题而异。

# 基于距离的分类方法

数据样本都是用n维数值属性描述的向量。

$$X=(x_1,x_2, \ldots\ldots x_n)$$

每个样本代表n维空间的一个点。 这样所有的训练样本都存放在n维模式空间中。

我们根据样本的m个类别，将同类样本计算出类中心：

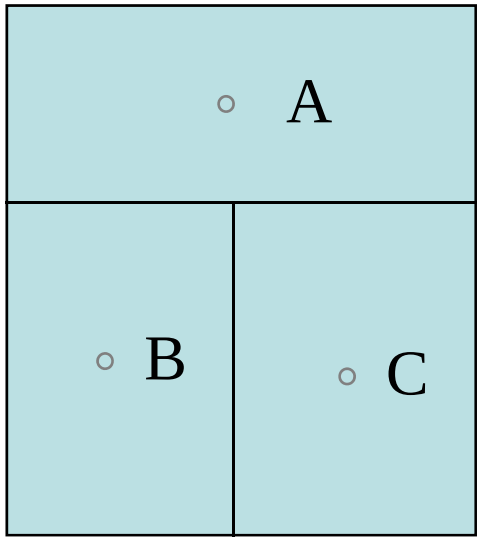
$$C_j=(c_{j1},c_{j2}, \ldots\ldots c_{jn}); \quad j=1,2,\ldots m$$

输入一个未知类别样本：

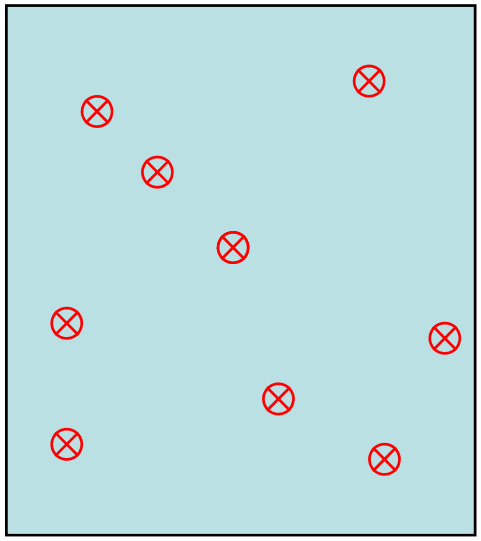
$$Y=(y1,y2, \ldots\ldots yn)$$

判断Y的类别，将Y与$C_j$进行距离计算，与那个类距离小，就是那类。计算距离方法因问题而异。

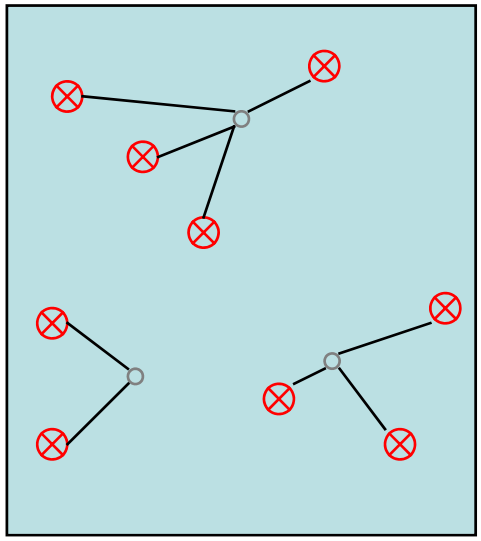
# 基于距离的分类方法




训练集（分3类）



9个未分类数据



将9个数据分类

# 基于距离的分类方法

输入：

C1,C2,…Cm //样本有m个类

t //未知样本

输出

C //t 属于的类

基于距离的算法

```
Dist=∞
For  i：=1 to m  do
   if dis（Ci，t）＜dist, then
   C= Ci
   dist=dis（Ci，t)
```

# K-NN算法例子

给样本数据集T={2, 4, 10, 12, 3, 20, 22, 21, 11, 24}
t={18}, K=4

1. N={2, 4, 10, 12}, d1=16, d2=14, d3=8, d4=6
2. d={3}, 比较, N={4, 10, 12, 3}, d1=14, d2=8, d3=6, d4=15
3. d={20}, 比较, N={10, 12, 3, 20}, d1=8, d2=6, d3=15, d4=2
4. d={22}, 比较, N={12, 3, 20, 22}, d1=6, d2=15, d3=2, d4=4
5. d={21}, 比较, N={3, 20, 22, 21}, d1=15, d2=2, d3=4, d4=3
6. d={11}, 比较, N={20, 22, 21, 11}, d1=2, d2=4, d3=3, d4=7
7. d={24}, 比较, N={20, 22, 21, 24}, d1=2, d2=4, d3=3, d4=6

t属于{20, 22, 21, 24}所在的类.

# K-NN算法例子2

给样本数据集:

T={(1,0),(1,2),(1,4),(2,1),(2,3),(3,1),(3,-3),(5,0),
(5,-1),(6,1)}

t={(4,2)},K=4

1. N={(1,0),(1,2),(1,4),(2,1)},
d1=sqr(13),d2=3,d3=sqr(13),d4=sqr(5)

2. d={(2,3)},dt=sqr(5),N={(1,2),(1,4),(2,1),(2,3)},
d1=3,d2=sqr(13),d3=sqr(5),d4=sqr(5)

3. d={(3,1)}, dt=sqr(2),N={(1,4),(2,1),(2,3),(3,1)},
d1=sqr(13),d2=sqr(5),d3=sqr(5),d4=sqr(2).

4. d={(3,-3)}, dt=sqr(26),N={(1,4),(2,1),(2,3),(3,1)},
d1=sqr(13),d2=sqr(5),d3=sqr(5),d4=sqr(2).

# K-NN算法例子2

5. d={(5, 0)}, dt=sqr(5), N={(2, 1), (2, 3), (3, 1), (5, 0)},
   d1=sqr(5), d2=sqr(5), d3=sqr(2), d4=sqr(5).
6. d={(5, -1)}, dt=sqr(10), N={(2, 1), (2, 3), (3, 1), (5, 0)},
   d1=sqr(5), d2=sqr(5), d3=sqr(2), d4=sqr(5).
7. d={(6, 1)}, dt=sqr(5), N={(2, 3), (3, 1), (5, 0), (6, 1)},
   d1=sqr(5), d2=sqr(2), d3=sqr(5), d4=sqr(5).

# 贝叶斯分类方法

# 贝叶斯方法产生和发展

- 起源：贝叶斯统计分析起源于1763 年Bayes的一篇论文
- 上世纪30年代，形成了贝叶斯学派。
- 上世纪50-60年代，发展成了一个很有影响的统计学派。
- 上世纪80年代，贝叶斯网络应用于专家系统，成为表示不确定性知识和推理的一种流行方法。
- 上世纪90年代，随着数据挖掘技术的出现和发展，贝叶斯网络开始用于数据挖掘任务。

# 贝叶斯分类方法

- 贝叶斯分类是统计学分类方法。该方法可以预测类成员关系的可能性。给一个样本，预测属于某个类的概率。
- 贝叶斯分类方法是基于贝叶斯定理。用朴素的贝叶斯分类可与决策树和神经元网络相媲美。
- 在大型数据库中它具有高准确度和高速度。
- 朴素的贝叶斯分类是类条件独立，而贝叶斯网络则属性间依赖。

# 贝叶斯分类方法

- 贝叶斯定理
- 朴素贝叶斯分类
- 贝叶斯网络

# 贝叶斯定理

- $X$：是一个未标注测试样本

  $X$ 是由一些属性表示，（但不知属于那类，要确定其类）

- $H$：是一个假设， 如假设$X$ 属于类C。
- $P(H/X)$：

  我们希望确定$X$ 条件下$H$ 成立的概率，X是给定观测样本（观测到一些属性），要确定$X$ 属于C的概率，用贝叶斯方法计算出来。这是$H$ 的后验概率。

- $P(H)$：$H$ 先验概率(任意一个样本属于C类的概率)
- $P(X)$：$X$ 先验概率，具有这些特征的样本，属于C类的概率
- $P(X/H)$：在$H$ 条件下，$X$成立的概率，这是X的后验概率。
- 贝叶斯定理为：$P(H|X)=\frac{P(X|H)P(H)}{P(X)}$

# 例子

- 假定数据的样本域为水果，它们用颜色和形状描述。

- 如苹果颜色为红色，形状为圆形。

X：是颜色为红色，形状为圆的物体,不知道是什么东西（不知属于哪类？）

H：X是苹果的假设

P(H|X)：在X是颜色为红色，形状为圆的物体条件下,H成立(即X为苹果)的概率;在X条件下,H的后验概率。

P(H)：先验概率，给任意一个物体为苹果的概率。

P(H|X)（后验概率）比 P(H)（ 先验概率）基于更多的背景知识（有更多信息）

P(X)：X先验概率，取出一个样本，其为红色且圆的概率。

P(X|H)：在X为苹果条件下（即在H成立下）,X颜色为红色，形状为圆的概率。在H条件下X的后验概率。

贝叶斯公式 $P(H|X)=\frac{P(X|H)P(H)}{P(X)}$ 给出了这些关系

在给定的数据集情况下：

P(H)：通过数据可以计算出来，一般为常数。

P(X)：通过数据可以计算出来，一般为常数。

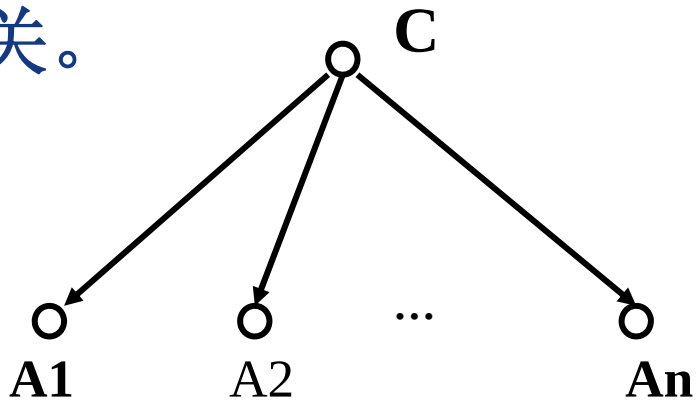
P(X|H)：通过数据可以计算出来

P(H|X)：由上述公式可计算出来。

# 朴素贝叶斯分类

1. 每个样本 $X$ 由n维向量表示 $X=\{x_1,x_2,\ldots\ldots x_n\}$,

它代表n个属性$A_1$, $A_2$,..... $A_n$ 一点取值， 如属性是相互独立,我们称其为朴素贝叶斯，说明它的属性间无关系,属性只与类有关。一个的属性取值与其它属性取值无关。




A1到An的取值都是独立的，属性A1取值与属性A2...An取值没有关系，它的取值也不影响其它属性取值。它们取值只影响类别C。

2. 假定有m个类 $C_1$， $C_2$，…… $C_m$，给一个测试样本 $X=\{x_1, x_2, \ldots\ldots x_n\}$，有n个属性， 不知道它属于那类，用贝叶斯方法，可求出X可能属于哪类 $C_i$ ，当且仅当：

$$P(C_i|X) > P(C_j|X) \quad 1 \leqq j \leqq m,\ j \neq 1$$

最大化 $P(C_i|X)$ ，最大的类$C_i$ 称最大后验假定。

根据贝叶斯定理 $$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

可以计算出最大的类$C_i$。

3. 最大化 $P(C_i|X)=\frac{P(X|C_i)P(C_i)}{P(X)}$

$P(X)$ 对所有类一般为常数

最大化 $P(X|C_i)P(C_i)$

有时 $P(C_i)$ 为等概率

只需最大化 $P(X|C_i)$

否则最大化 $P(X|C_i)P(C_i)$

$P(X|C_i)$ 和 $P(C_i)$ 都可由数据计算出来。

4. 利用样本集可分别计算 $P(X|C_i)$ 和 $P(C_i)$

在属性独立的情况下，就是朴素贝叶斯的情况。

- $P(X|C_i)$ 可用下式代替

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

其中 $P(x_k|C_i) = S_{ik} / S_i$

$S_i$ 是样本集中为$C_i$类的个数，而$S_{ik}$是第$k$个属性取值$x_k$为$C_i$类的个数， $C_i$类取值$x_k$的概率就是$S_{ik}/S_i$

设总样本集为S，有m个类，每类分别有样本S1,S2,…Sm, 这样，第Ci类的概率为：

$$P(C_i) = s_i / S$$

5. 测试样本X对应利用公式计算出的最大概率的类

$$P(C_i | X) = \frac{P(X | C_i) P(C_i)}{P(X)}$$

# 举例1

**Y=(age ≤30,income="medium",student="yes", credit_rating="fair")**

**Y 属于那类？ 即 buys_computer=? （yes还是no）**

| A1 | A2 | A3 | A4 | C |
|---|---|---|---|---|
| age | income | student | credit_rating | buys_computer |
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

# 利用贝叶斯公式计算

$$P(Y|C_i)P(C_i)$$

先计算

$$P(C_i)$$

再计算

$$P(Y|c_j)=\prod_{i=1}^{n}P(y_i|c_j)$$

**C 仅 为 2 类 C1: buys_computer=yes，9**

**C2: buys_computer=no， 5**

计算 P(C*i*)

P(C1)=p(buys_computer=yes)=9/14= **0.643**

P(C2)=p(buys_computer=no )=5/14= **0.357**

计算 $$\mathbf{P(Y \mid c_j) = \prod_{i=1}^{n} P(y_i \mid c_j)}$$

样本Y的属性为4个（n=4），分别为//8次计算

y1=age ≤”**30**” , **y2= income“medium”,**

**y3=student=“yes”, y4=credit_rating =“fair”**

P ( age ≤”**30**”| **buys_computer=“yes”)=2/9=0.222**

P ( age ≤”**30**”| **buys_computer=“no”)= 3/5=0.6**

- P( income="**medium**" | **buys_computer="yes")=4/9=0.444**
- P( income="**medium**" | **buys_computer="no")=2/5=0.2**
- P( student="**yes**" | **buys_computer="yes")=6/9=0.667**
- P( student="**yes**" | **buys_computer="no")=1/5=0.2**
- P( credit="**fair**" | **buys_computer="yes")=6/9=0.667**
- P( credit="**fair**" | **buys_computer="no")=2/5=0.2**

**使用以上概率，得：**

- P(Y | **buys_computer="yes")**
  **=0.222x0.444x0.667x0.667=0.044**
- P(Y | **buys_computer="no") =0.6x0.4x0.2x0.4=0.019**
- P( **buys_computer="yes")=0.044x0.643=0.028**
- P( **buys_computer="no")=0.019x0.357=0.007**

$$P(Y|C_1)P(C_1) \qquad P(Y|C_2)P(C_2)$$

因此，对于样本Y，

Y=(age ≤30,income="medium",student="yes",
credit_rating="fair"）

朴素贝叶斯分类预测结论

$$P(C1 \mid Y) > P(C_2 \mid Y)$$

因而，样本Y为 C1类：buys_computer="yes"

# 举例2

| Name | Give Birth | Can Fly | Live in Water | Have Legs | Class |
|---|---|---|---|---|---|
| human | yes | no | no | yes | mammals |
| python | no | no | no | no | non-mammals |
| salmon | no | no | yes | no | non-mammals |
| whale | yes | no | yes | no | mammals |
| frog | no | no | sometimes | yes | non-mammals |
| komodo | no | no | no | yes | non-mammals |
| bat | yes | yes | no | yes | mammals |
| pigeon | no | yes | no | yes | non-mammals |
| cat | yes | no | no | yes | mammals |
| leopard shark | yes | no | yes | no | non-mammals |
| turtle | no | no | sometimes | yes | non-mammals |
| penguin | no | no | sometimes | yes | non-mammals |
| porcupine | yes | no | no | yes | mammals |
| eel | no | no | yes | no | non-mammals |
| salamander | no | no | sometimes | yes | non-mammals |
| gila monster | no | no | no | yes | non-mammals |
| platypus | no | no | no | yes | mammals |
| owl | no | yes | no | yes | non-mammals |
| dolphin | yes | no | yes | no | mammals |
| eagle | no | yes | no | yes | non-mammals |

| Give Birth | Can Fly | Live in Water | Have Legs | Class |
|---|---|---|---|---|
| yes | no | yes | no | ? |

# 属性与类

A: attributes

M: mammals

N: non-mammals

$P(A|M)=6/7*6/7*2/7*2/7=0.06$

$P(A|N)=1/13*10/13*3/13*4/13=0.0042$

$P(M)=7/20,\ P(N)=13/20$

$P(A|M)*P(M)=0.021 > P(A|N)*P(N)=0.0027$

➔ $A \in$ Mammals

# Thank you!