最近邻学习方法

6.1　最近邻决策规则

6.2　剪辑最近邻法

6.3　压缩最近邻法

最近邻学习的工作机制

k最近邻(k-Nearest Neighbor, kNN)学习是一种常用的有监督学习方法:

- 确定训练样本，以及某种距离度量
- 对于某个给定的测试样本，找到训练集中距离最近的k个样本，对于分类问题使用“投票法”获得预测结果，对于回归问题使用“平均法”获得预测结果。还可基于距离远近进行加权平均或加权投票，距离越近的样本权重越大
  - 投票法：选择这k个样本中出现最多的类别标记作为预测结果
  - 平均法：将这k个样本的实值输出标记的平均值作为预测结果

最近邻学习的工作机制

最近邻学习没有显式的训练过程，属于“懒惰学习”

- “懒惰学习”(lazy learning): 此类学习技术在训练阶段仅仅是把样本保存起来，训练时间开销为零，待收到测试样本后再进行处理

- “急切学习”(eager learning): 在训练阶段就对样本进行学习处理的方法

6.1 最近邻决策规则—1-NN

c类问题, 设 $\vec{x}_j^{(i)} \in \omega_i \quad (i=1,2,\cdots c \ , j=1,2,\cdots N_i)$

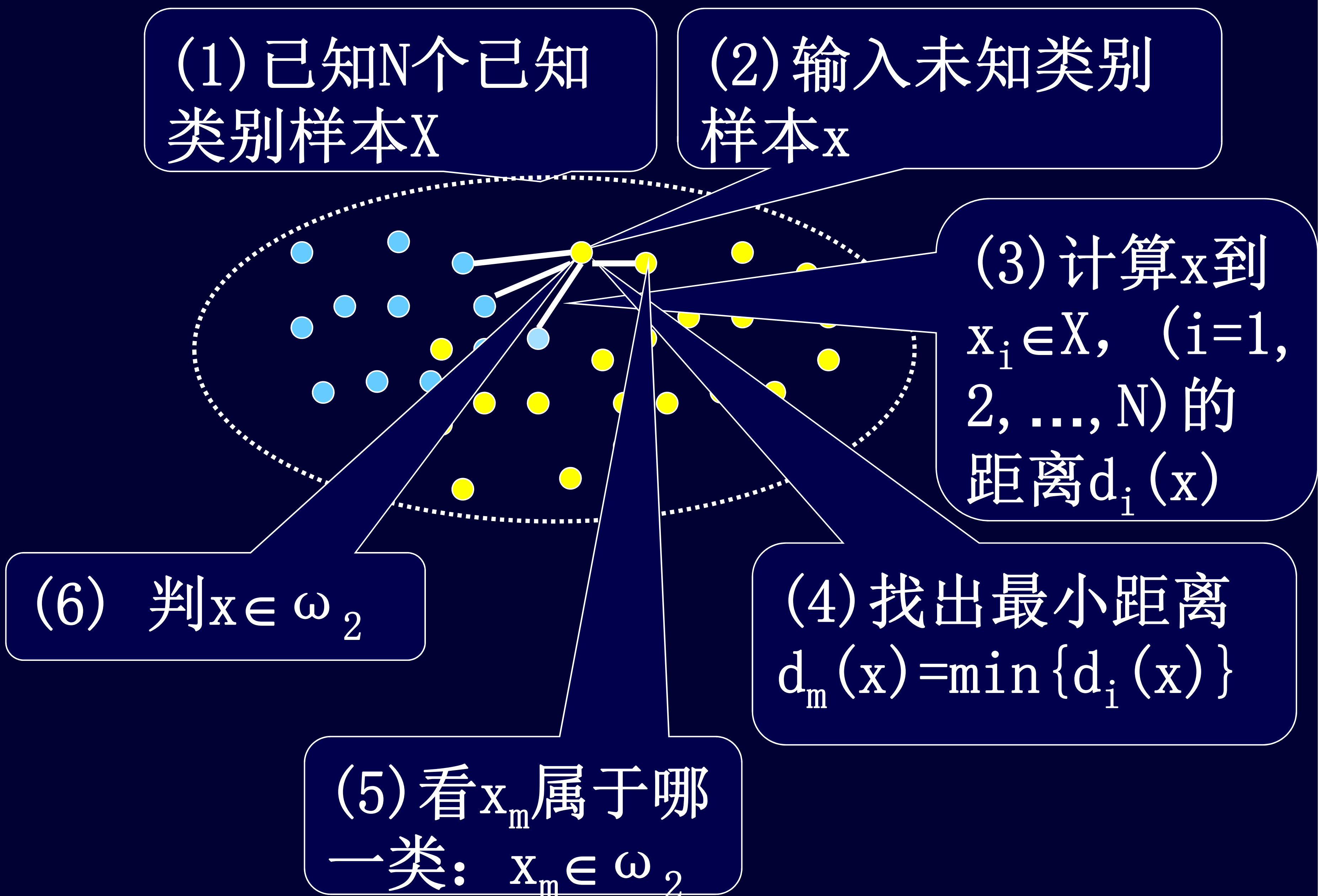
最近邻分类规则:

对待识别样本 $\vec{x}$ ，分别计算它与 $N=\sum_{i=1}^{c} N_i$ 个已知类别的样本 $\vec{x}_j^{(i)}$ 的距离, 将它判为距离最近的那个样本所属的类。

即 $$d_i(\vec{x}) = \underbrace{\min}_{j=1,2,\cdots,N_i} \left\| \vec{x} - \vec{x}_j^{(i)} \right\| \quad i=1,2,\cdots,c$$

如果 $d_m(\vec{x}) = \underbrace{\min}_{i=1,2,\cdots,c} d_i(\vec{x})$ 则 $\vec{x} \in \omega_m$

6.1 最近邻决策规则—1-NN



6.1 最近邻决策规则—k-NN

k-NN分类思想:

对待识别样本 $\vec{x}$ ，分别计算它与 $N=\sum_{i=1}^{c} N_i$ 个已知类别的样本 $\vec{x}_j^{(i)}$ 的距离，取k个最近邻样本，这k个样本中哪一类最多，就判属哪一类。

即，令 $\vec{x}$ 与 ω_i 的距离 $d_i(\vec{x})=k_i \quad i=1,2,\cdots,c; \quad \sum_{i=1}^{c} k_i = k$

如果 $d_m(\vec{x}) = \underbrace{\max}_{i=1,2,\cdots,c} d_i(\vec{x})$ 则 $\vec{x} \in \omega_m$

其中 k_i 表示 k 个近邻元中属于 ω_i 的样本个数

6.1 最近邻决策规则—k-NN

(1) 已知N个已知类别样本X

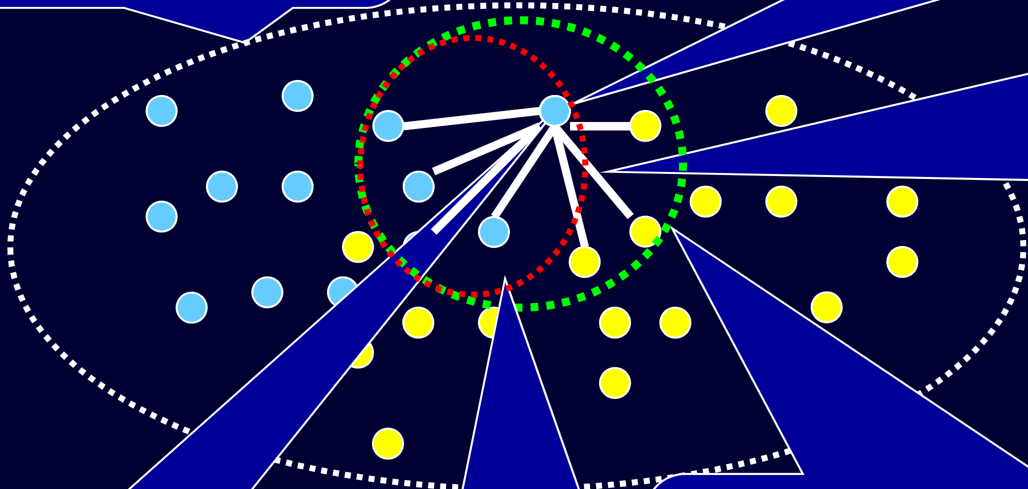
(2) 输入未知类别样本x

(3) 计算x到$x_i \in X$，$(i=1, 2, \ldots, N)$的距离$d_i(x)$

(4) 找出x的k个最近邻元$X_k=\{x_i, i=1, 2, \ldots, k\}$

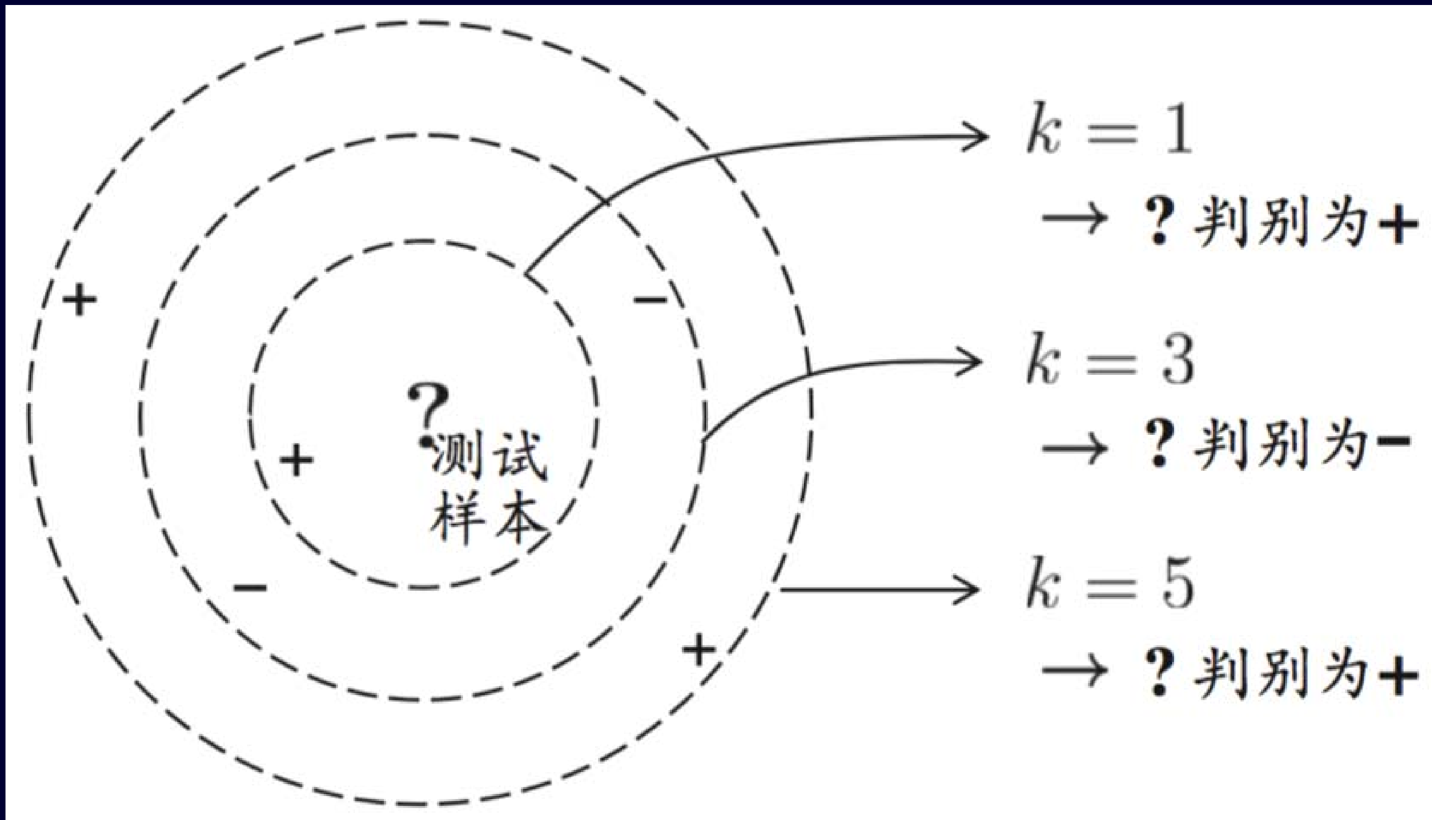
(5) 看X_k中属于哪一类的样本最多$k_1=3<k_2=4$

(6) 判$x \in \omega_2$



6.1 最近邻决策规则—k-NN

最近邻决策规则中的k是一个重要参数，当k取不同值时，分类结果会有显著不同



6.2 剪辑最近邻法

基本思想

- 一个现象:
  - 当不同类别的样本在分布上有交迭部分的，分类的错误率主要来自处于交迭区中的样本
- 当我们得到一个作为识别用的参考样本集时，由于不同类别交迭区域中不同类别的样本彼此穿插，导致用近邻法分类出错
- 因此如果能将不同类别交界处的样本以适当方式筛选，可以实现既减少样本数又提高正确识别率的双重目的
- 为此可以利用现有样本集对其自身进行剪辑

6.2 剪辑最近邻法

对于两类问题，设将已知类别的样本集$X^{(N)}$分成参照集$X^{(NR)}$和测试集$X^{(NT)}$两部分，$X^{(NR)} \cap X^{(NT)}=\Phi$，它们的样本数各为NR和NT，NR+NT=N。利用参照集$X^{(NR)}$中的样本 $\vec{y}_1, \vec{y}_2, \cdots, \vec{y}_{NR}$ 采用最近邻规则对已知类别的测试集$X^{(NT)}$中的每个样本 $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_{NT}$ 进行分类，剪辑掉$X^{(NT)}$中被错误分类的样本。

若$\vec{y}^0(\vec{x}) \in X^{(NR)}$是$\vec{x} \in X^{(NT)}$的最近邻元，剪辑掉与$\vec{y}^0(\vec{x})$异类的 $\vec{x}$ ，余下的判决正确的样本组成剪辑样本集$X^{(NTE)}$ 。这一操作称为**剪辑**。

剪辑最近邻法

获得剪辑样本集$X^{(NTE)}$后，对待识样本 $\vec{x}$ 采用最近邻规则进行分类。

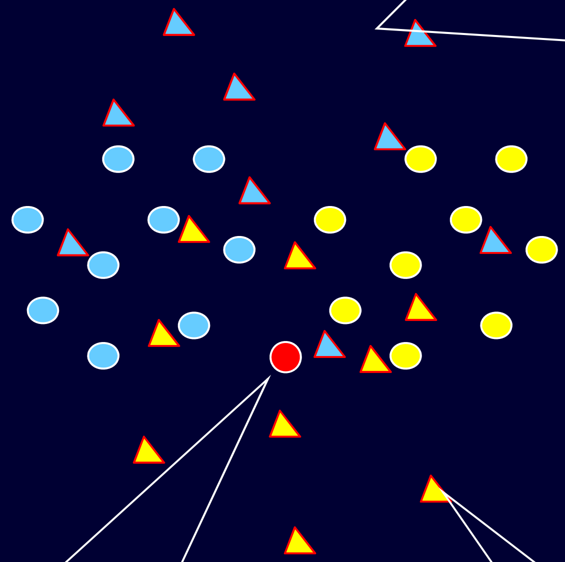
$$d_i(\vec{x}) = \underbrace{\min}_{j=1,2,\cdots,N_i} \left\| \vec{x} - \vec{x}_j^{(i)} \right\| \quad i = 1, 2, \cdots, c$$

如果 $d_m(\vec{x}) = \underbrace{\min}_{i=1,2,\cdots,c} d_i(\vec{x})$ 则 $\vec{x} \in \omega_m$

这里 $\vec{x}_j \in X^{(NTE)}$

剪辑最近邻法

- ■ $\in\omega_1$
- ■ $\in\omega_2$
- ○ $\in X^{(NR)}$
- △ $\in X^{(NT)}$



用$X^{(NR)}$中的样本采用最近邻规则对$X^{(NT)}$中的每个样本分类，剪辑掉$X^{(NT)}$中被错误分类的样本

余下判决正确的样本组成剪辑样本集$X^{(NTE)}$

用$X^{(NTE)}$ 对输入的未知样本做K-NN分类

剪辑最近邻法

6.2.2 剪辑k-NN最近邻法

剪辑最近邻法可以推广至k-NN近邻法中。步骤：

第一步 用k-NN 法进行剪辑；

第二步 用1-NN 法进行分类。

如果样本足够多，就可以重复地执行剪辑程序，以进一步提高分类性能。称为重复剪辑最近邻法。

6.2.3 重复剪辑最近邻法

MULTIEDIT算法

（1） 将样本集$X^{(N)}$随机地划分为s个子集：

$$X^{(N)} = \{X_1, X_2, \cdots, X_s\} \qquad (s \geq 3)$$

（2） 用最近邻法，以$X_{(i+1) \bmod s}$为参照集，对X_i中的样本进行分类，其中i =1, 2, …, s；

（3）去掉（2）中被错误分类的样本；

（4）用所留下的样本构成新的样本集$X^{(NE)}$；

（5）如果经过k 次迭代再没有样本被剪辑掉则停止；否则转至（1）。

6.3 压缩最近邻法

- 远离分类边界的样本对于分类决策没有贡献
- 压缩近邻法：利用现有样本集，逐渐生成一个新的样本集，使该样本集在保留最少量样本的条件下，仍能对原有样本的全部用最近邻法正确分类，那么该样本集也就能对待识别样本进行分类，并保持正常识别率

6.3 压缩最近邻法

- 定义两个存储器，一个用来存放即将生成的样本集，称为Store；另一存储器则存放原样本集，称为Grabbag

1. **初始化**。Store是空集，原样本集存入Grabbag；从Grabbag中任意选择一样本放入Store中作为新样本集的第一个样本
2. **样本集生成**。在Grabbag中取出第i个样本用Store中的当前样本集按最近邻法分类。若分类错误，则将该样本从Grabbag转入Store中，若分类正确，则将该样本放回Grabbag中
3. **结束过程**。若Grabbag中所有样本在执行第二步时没有发生转入Store的现象，或Grabbag已成空集，则算法终止，否则转入第二步

最后以Store中的样本 作为最近邻法的设计集