第五章 近邻法

是一种分段线性分类器

直接根据训练样本对新样本进行分类

距离度量

度量$D(\cdot,\cdot)$本质上是一个函数，该函数给出了两个模式之间的标量距离的大小。一个度量必须满足4个性质：

对于任意的向量a，b，和c，有

- 非负性：$D(a,b) \geq 0$
- 自反性：$D(a,b)=0$ 当且仅当 $a=b$
- 对称性：$D(a,b)=D(b,a)$
- 三角不等式$D(a,b)+D(b,c) \geq D(a,c)$

距离度量

*d*维空间中的欧式距离

$$D(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} (a_k - b_k)^2 \right)^{1/2}$$

能够满足这些性质

距离度量

更为一般的*d*维空间的度量为Minkowski距离度量

$$L_k(\mathbf{a},\mathbf{b}) = \left(\sum_{i=1}^{d} | a_i - b_i |^k \right)^{1/k}$$

通常也被称为L_k范数

欧式距离就是L_2范数

距离度量

L_1范数

$$L_1(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{d} |a_i - b_i|$$

也被称为Manhattan距离或街区距离、绝对距离

显然，欧式距离和绝对距离是明氏距离的两个特例

手工运算时，为简便起见，通常采用绝对距离

最近邻法

1．最近邻法规则

已知C类，每类样本数为Ni个，$i = 1, 2, \ldots c$

判别函数：$g_i(x) = \min \left\| x - x_i^k \right\| \quad k = 1, 2, \ldots, N_i$

决策规则：$g_j(x) = \min_i g_i(x) \quad i = 1, 2, \ldots c$

则决策 $x \in w_j$

——称为最近邻法

昔孟母，择邻处

最近邻法实质：

就是将样本x与N个已知类别属性的样本之间的欧氏距离进行比较，将x归入最近的样本所属的类别。

最近邻法是次优方法，误差率比贝叶斯误差率大，当N→∞，误差率不超过贝叶斯误差率2倍。

但不具有统计特性，不稳定。

- 在样本数N很大时，最近邻规则能很好的工作

∵样本数非常大时，认为x距离x′足够近，使

$$P(w_i / x') = P(w_i / x)$$

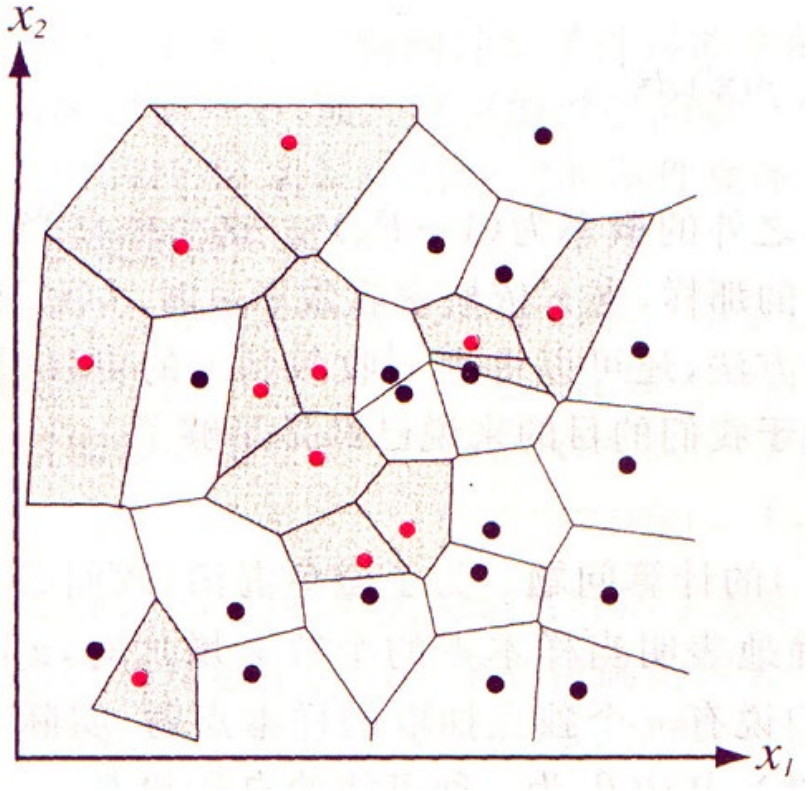
∴ 最近邻规则是真实概率的一个有效近似

相当于决策规则为：

$$P(w_i \mid x) = \max_j P(w_j \mid x) \qquad x \in w_i$$

最近邻规则相当于把特征空间分成一个个网格单元，每一个单元的点到最近邻x′（代表点）的距离比到其它样本点距离更近

∴小单元中的任意点的类别就与最近邻x′的类别相同。即x与N个训练样本比较欧氏距离，x归入最近样本的类。



如图示二维情况，分界面就是各相邻训练样本距离的垂直中心线

∴ 最近邻决策面是分段线性的。

最近邻法效果分析：

通过求无限样本下的平均条件错误率p(e|x)进行分析（略）

可证明，存在下列关系：

$$P^* \leqslant P \leqslant P^*(2-\frac{c}{c-1}P^*)$$

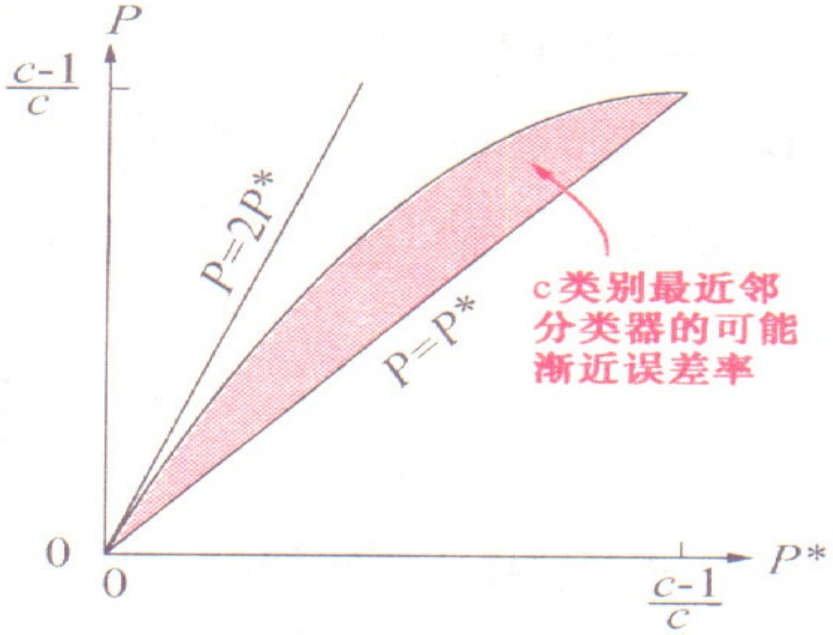
其中：

P为无限样本数时的最近邻法的错误率

P*为贝叶斯错误率，即最小错误率，介于 $0 \sim \frac{c-1}{c}$ 之间。

（P(w/x)=0时， P* ＝0； P(w/x)相等时， $P^* = \frac{c-1}{c}$）

最近邻法错误率的上下界与贝叶斯错误率的关系



可见，P总是小于等于2P*，当贝叶斯错误率P*很小时，P=2P*。当取极端情况时，上下界重合。

在N有限时，最近邻法效果如何？若没有关于概率分布的其它知识，很难有结论。

K—近邻法

基本规则：

找出x的k个近邻，k个近邻中多数属于哪一类，就把x归为哪一类。

x的k个近邻中，w_1中有k_1个，w_2中有k_2个，…，w_c中有k_c个，则

判别函数为：$g_i(x) = k_i \quad i = 1, 2, ..., c$

决策规则：$g_j(x) = \max_i k_i \qquad i = 1, 2, ..., c$

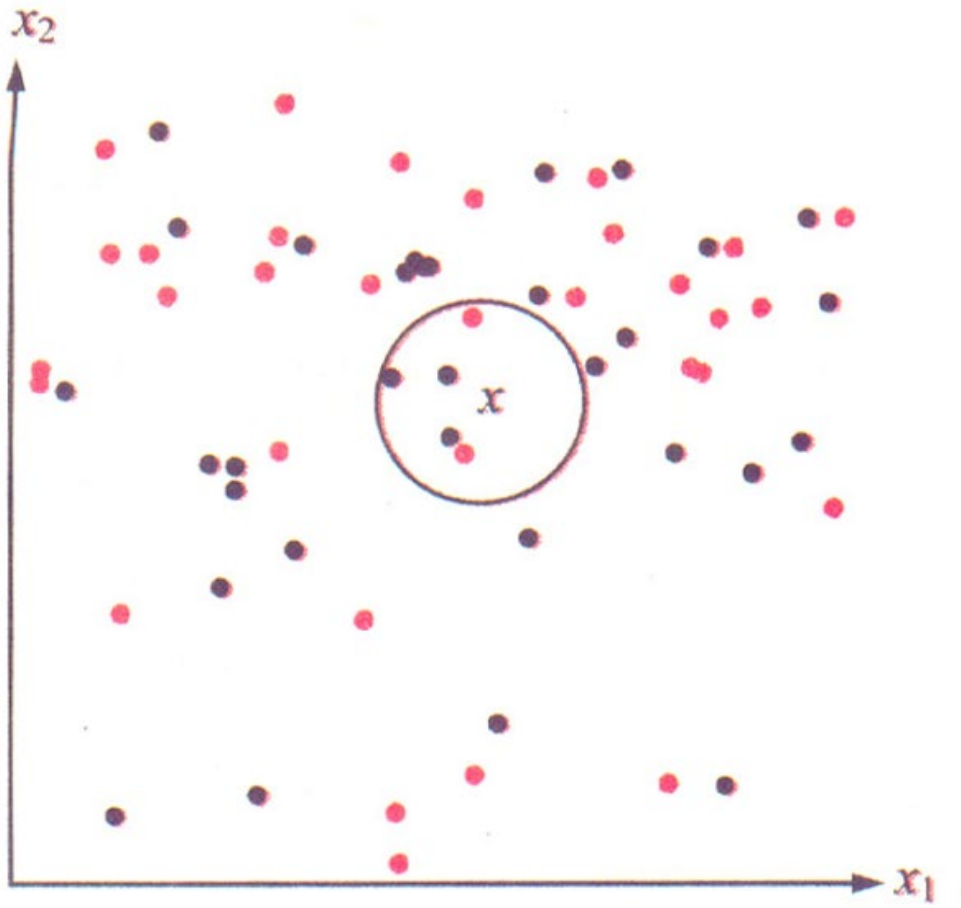
则决策 $x \in w_j$

出淤泥而不染是少数

例：

图中为k＝5的情况

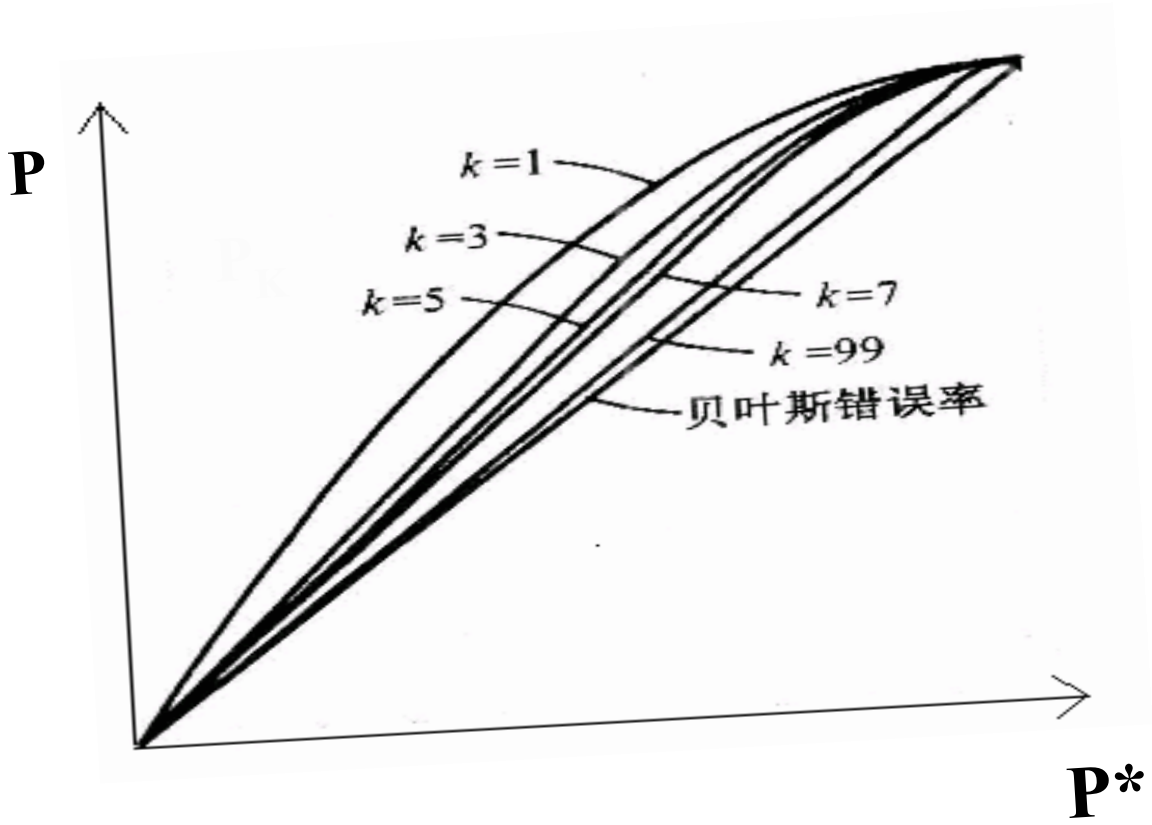
∴x归为黑色点所属类别



可证明，k一近邻法错误率P满足下列关系：

$$P^* \leqslant P \leqslant C_k(P^*)$$

k一近邻法错误率上下界与贝叶斯错误率的关系



$\therefore$ $k \to \infty$ $P = P^*$，k近邻法成为最优分类规则，即当N→∞才能保证k近邻法几乎是最优分类规则。

近邻法缺点：计算量大

➢近邻法的计算复杂度：　（空间复杂度和时间复杂度）

以最近邻法为例，其复杂度有较多的研究

设d维空间，N个已知训练样本

在最简单方法中，就是搜索每一个样本点，计算距离，找出距离最近的那一个。→主要是时间复杂度。

另一方法，并行实现方法。能保证搜索时间为常数，将x输入每一盒子中。 → 空间复杂度大。

近邻法的改进

- 快速算法
- 剪辑近邻法
- 压缩近邻法

近邻法的改进

快速算法

基本思想是将样本分级，分成一些不相交的子集，并在子集的基础上进行搜索

- 把样本集分级分成多个子集（树状结构）
- 每个子集（结点）可用较少几个量代表
- 通过将新样本与各结点比较排除大量候选样本
- 只有最后的结点（子集）中逐个样本比较，找出近邻

近邻法的改进

令$X=\{x_1, x_2, \ldots, x_N\}$表示样本集，我们的目的是在$X$中寻找样本x的$k$个近邻，为简单起见，先看最近邻的情况($k=1$)。

算法分为两个阶段：

- 将X分级分解
- 用搜索算法找出x的最近邻

近邻法的改进

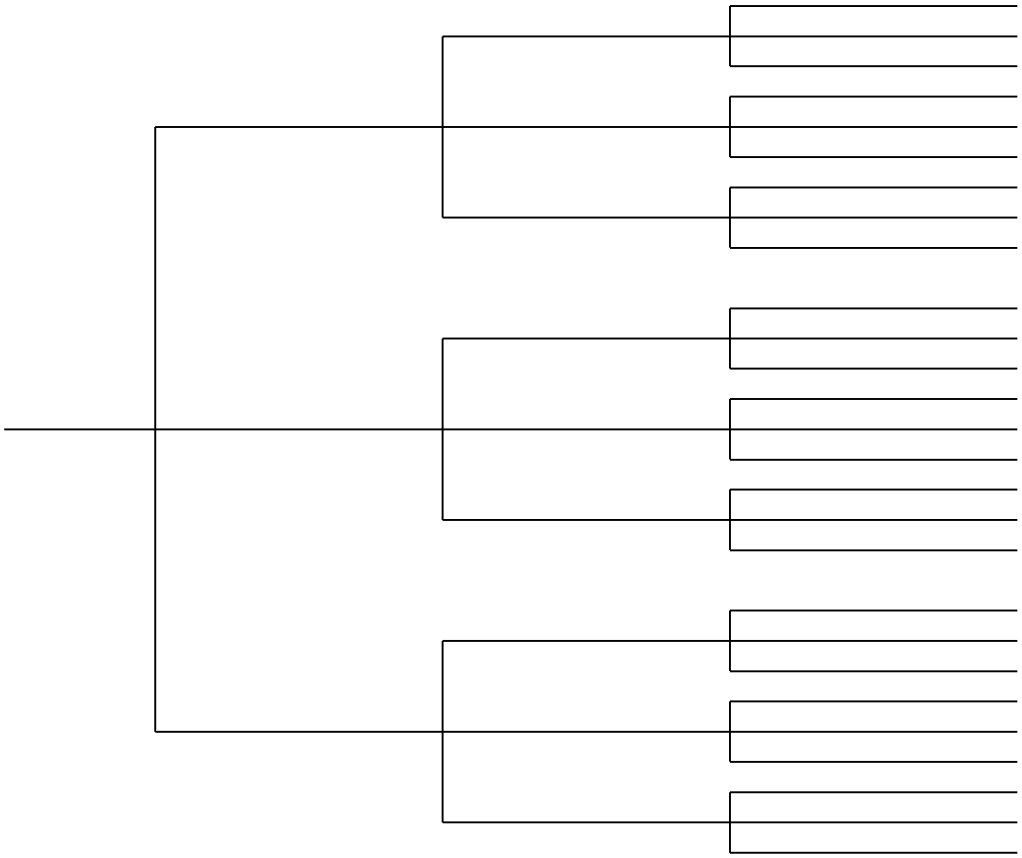
第一阶段：样本集分级分解

首先将X分为l个子集，每个子集再分成l个子集。

依次进行下去，就可以得到一个树结构。

近邻法的改进

分级示意图 L=0 L=1 L=2 L=3



近邻法的改进

令：

X_p：结点p对应的样本子集

N_p：X_p中的样本数

M_p：样本子集X_p中的样本均值

r_p：从M_p到$x_i \in X_p$的最大距离

近邻法的改进

第二阶段：搜索

首先给出两个规则，利用它们可以检验x的最近邻是否在X_p中。

规则1：如果存在

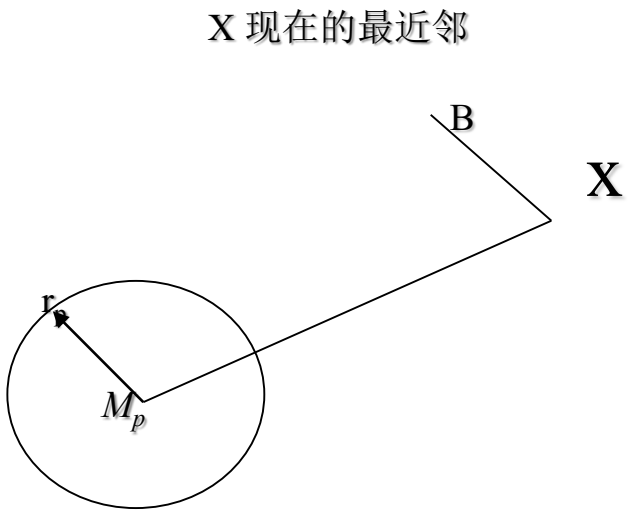
$$B + r_p < D(x, M_p)$$

则 $x_i \in X_p$不可能是x的最近邻

规则2：如果

$$B + D(x_i, M_p) < D(x, M_p)$$

其中$x_i \in X_p$，则x_i不是x的最近邻



近邻法的改进

树搜索算法

1. 置$B=\infty$，$L=0$，$p=0$（L是当前水平，p是当前结点）
2. 将当前结点的所有直接后继结点放入一个目录表中，并对这些结点计算$D(\mathbf{x}, \mathbf{M}_p)$
3. 对步骤2中的每个结点p，根据规则1，如果有

$$D(\mathbf{x}, \mathbf{M}_p) > B + r_p ,$$

则从目录表中去掉p

近邻法的改进

4. 如果步骤3的目录表中已经没有结点，则后退到前一个水平，即置$L=L-1$。如果$L=0$则停止，否则转步骤3。如果目录表中有一个以上的结点存在，则转步骤5

5. 在目录表中选择最近结点p′，它使$D(x, M_p)$最小化，并称该p′为当前执行结点，从目录表中去掉p′。如果当前的水平L是最终水平，则转步骤6。否则置$L=L+1$，转步骤2

近邻法的改进

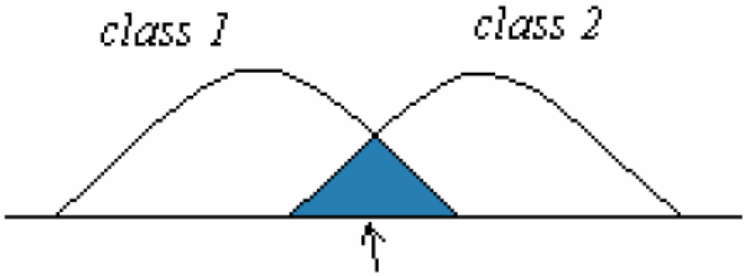
6. 对现在的执行结点p′中的每个x_i，利用规则2做如下检验，如果 $D(x, M_p) > B+D(x_i, M_p)$，则$x_i$不是x的最近邻，从而不计算$D(x, x_i)$，否则计算$D(x, x_i)$。若$D(x, x_i) < B$, 置$NN=i$和$B=D(x, x_i)$。在当前执行结点中所有$x_i$被检查之后，转步骤3

当算法结束时，输出x的最近邻 x_{NN} 和 x 与 x_{NN} 的距离 $D(x, x_{NN}) = B$

近邻法的改进

剪辑近邻法

基本思想：处在两类交界处或分布重合区的样本可能误导近邻法决策，应将它们从样本集中去掉。



近邻法的改进

基本方法

- 将样本集X^N分为考试集X^{NT}和参考集X^{NR}

$$X^N = X^{NT} \cup X^{NR}$$

$$X^{NT} \cap X^{NR} = \varnothing$$

- 剪辑：用X^{NR}中的样本对X^{NT}中的样本进行近邻法分类，剪掉X^{NT}中被错分的样本， X^{NT}中剩余样本构成剪辑样本集X^{NTE}
- 分类：利用X^{NTE}和近邻法对未知样本x 分类。

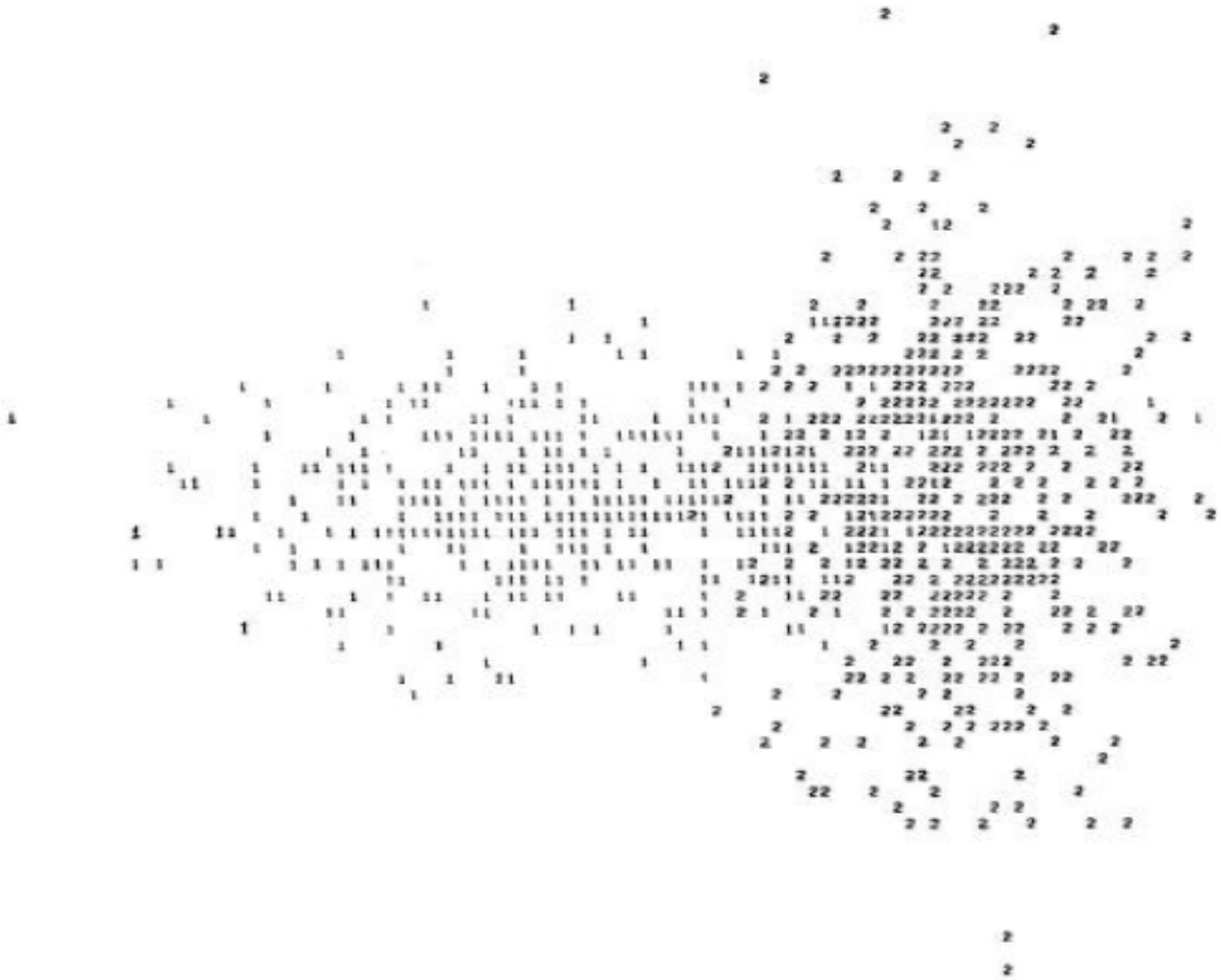
近邻法的改进

重复剪辑近邻法——MultiEdit算法

1. 将样本集X^N随机划分为s个子集，即

$$X^N = \{X_1, X_2, \ldots, X_s\} \text{ 且} s \geq 3$$

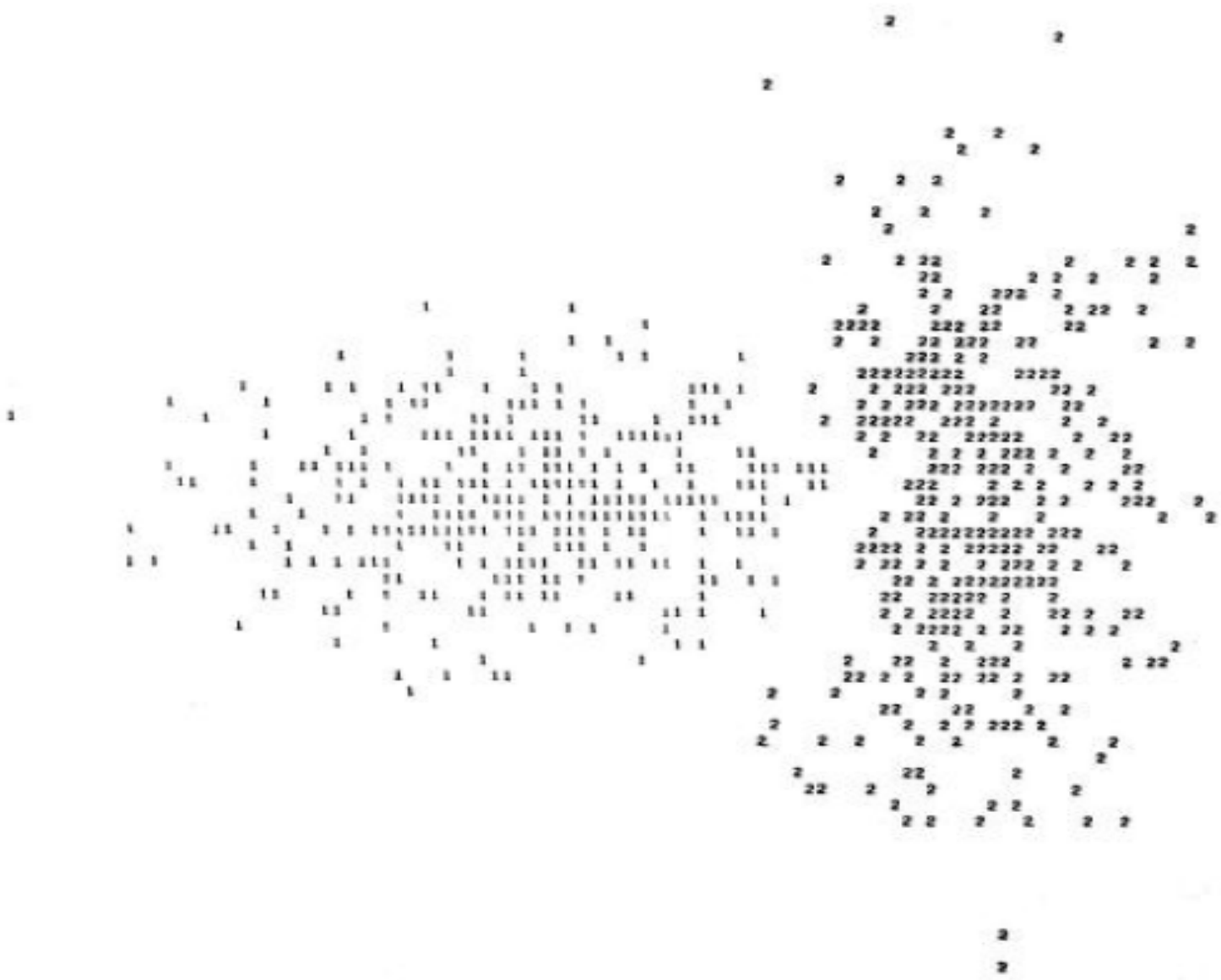
2. 用最近邻法，以$X_{(i+1)Mod(s)}$为参考集，对X_i中的样本进行分类，其中$i=1,2,\ldots,s$, $(i+1)Mod(s)$表示$i+1$对s求余
3. 去掉在2中被错分的样本
4. 用所有留下的样本，构成新的样本集X^{NE}
5. 如果经k次迭代，再没有样本被剪辑掉则停止，否则转1



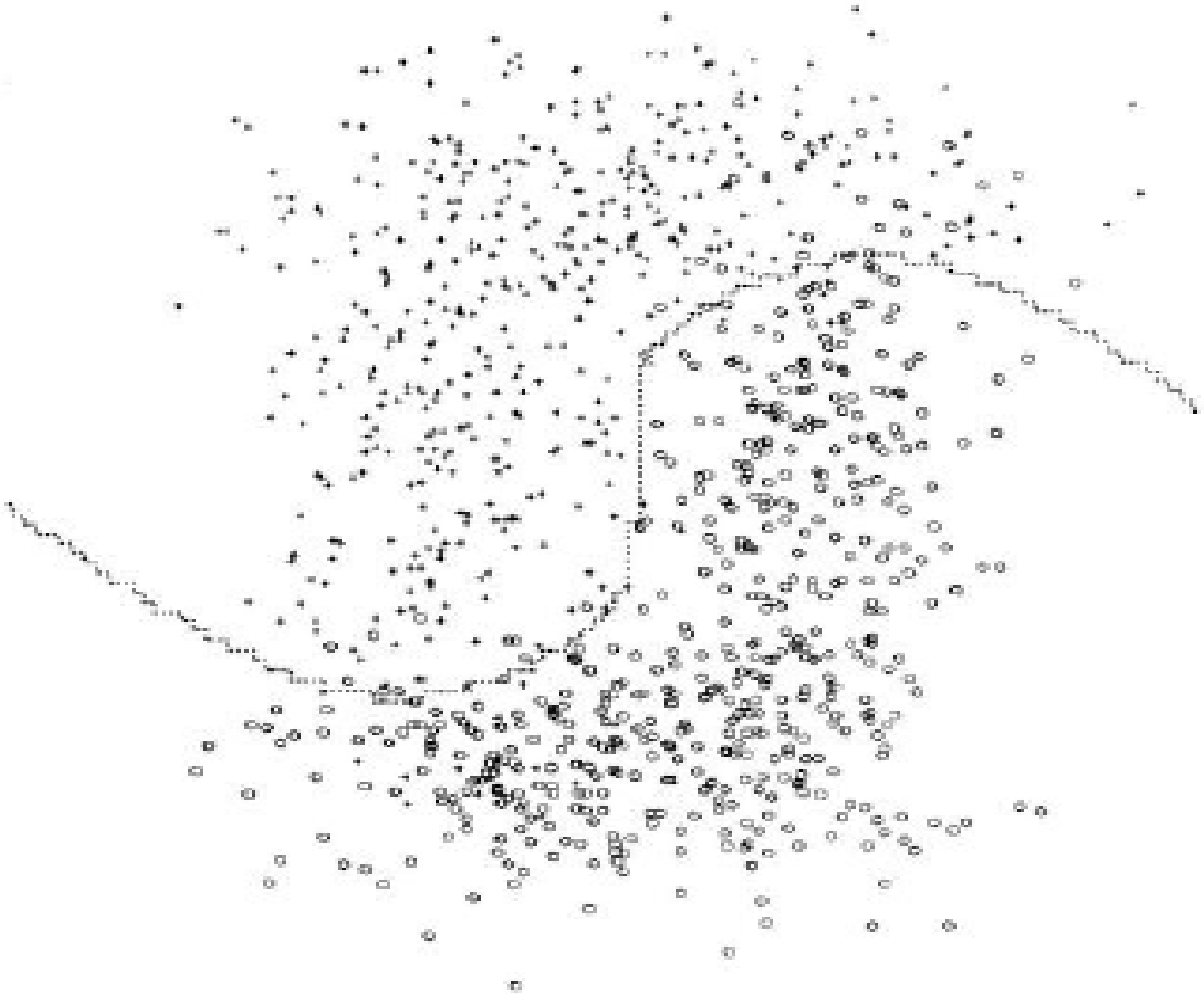
原始样本集

第一次迭代后留下的样本

第三次迭代后留下的样本

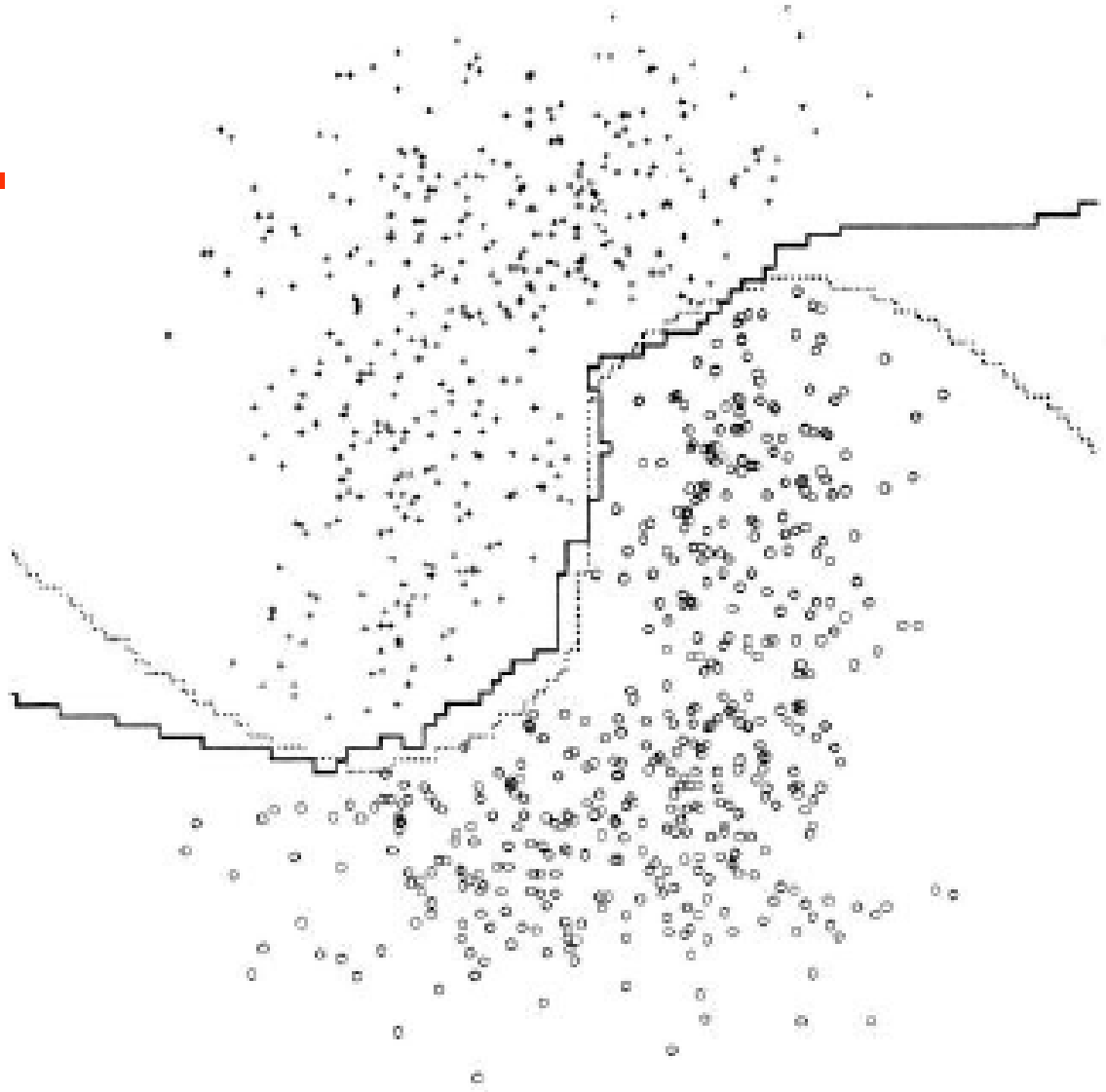


算法终止时留下的样本



初始样本集

第一次剪辑后的样本集



最终结果

近邻法的改进

- 压缩近邻法

剪辑的结果只是去掉了两类边界附近的样本，而靠近两类中心的样本几乎没有去掉。按照近邻规则，这些样本中的绝大多数对分类没有什么用处。因此在剪辑的基础上，再去掉一部分这样的样本有助于进一步缩短计算时间和减少存储量。一般称这类方法为压缩近邻法。

近邻法的改进

基本方法

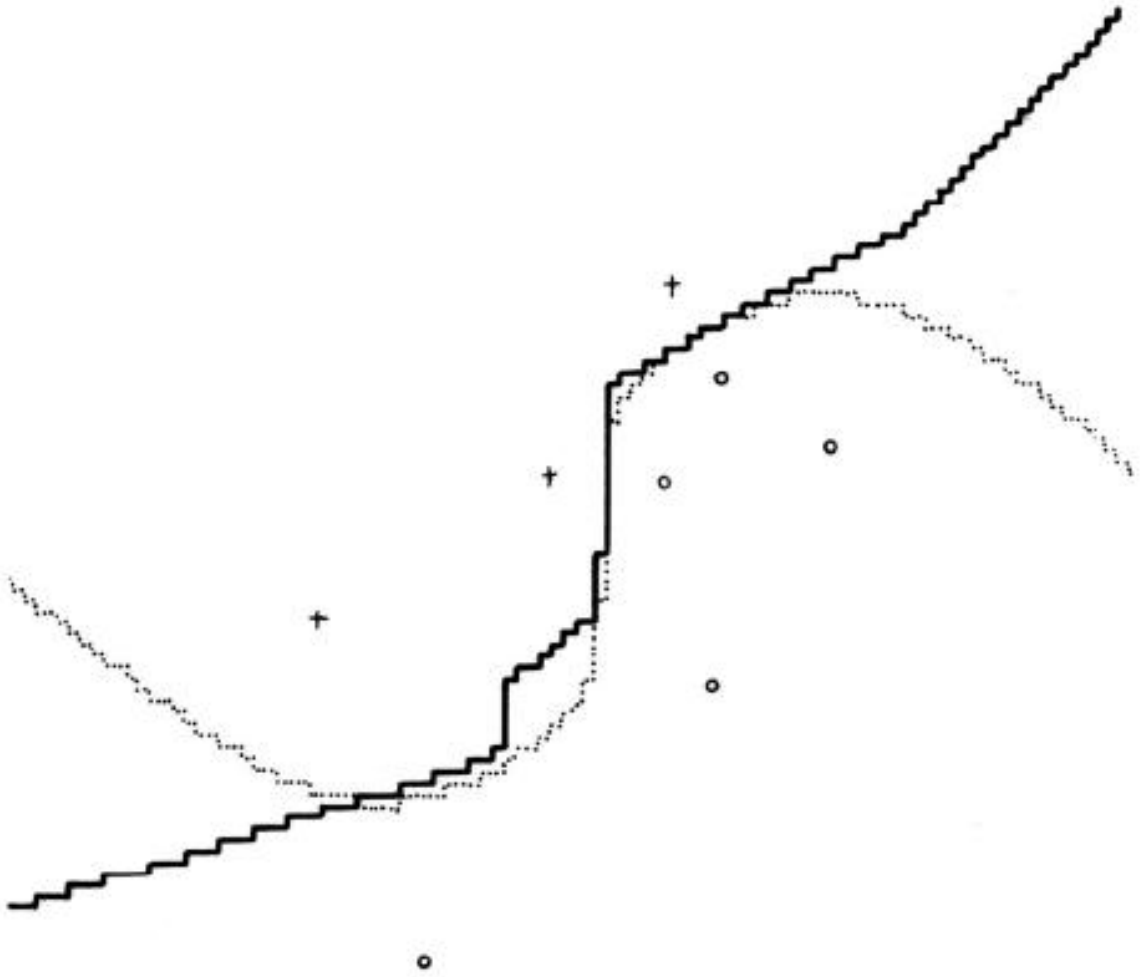
- 将样本集X^N分为X_S和X_G，开始时X_S中只有一个样本，X_G中为其余样本
- 考查X_G中每个样本，若用X_S可正确分类则保留，否则移入X_S
- 最后用X_S作最近邻法的设计集。

近邻法的改进

算法步骤（Condensing算法）

1. 设置两个存储器，分别为STORE和GRABBAG，将第一个样本放入STORE中，把其他样本放入GRABBAG中
2. 用当前STORE中的样本以最近邻规则对GRABBAG中的第*i*个样本进行分类。若分类正确，则该样本仍送回GRABBAG中，否则放入STORE中，对GRABBAG中所有样本重复上述过程
3. 若GRABBAG中的所有样本在进行上述检验过程中没有一个样本从GRABBAG转到STORE或者GRABBAG为空时，算法终止，否则转2

最后我们以STORE中的样本作为最近邻的设计集。



数据经MultiEdit算法剪辑后再使用Condensing压缩近邻算法的结果

近邻法的改进

可见，经压缩后，虽然误差稍大，但样本数目却大大减少，因此可以大大节省存储量和计算量。

可作拒绝决策的近邻法

具有拒绝决策的两类*k*-近邻法

确定

$$k' > \frac{k+1}{2}$$

若x的*k*个近邻中有大于或等于k′个属于某一类ω_i (*i*=*1*,*2*)，则决策

$x \in \omega_i$ ，否则就做拒绝决策。

简单多数 ⇒ 绝对多数