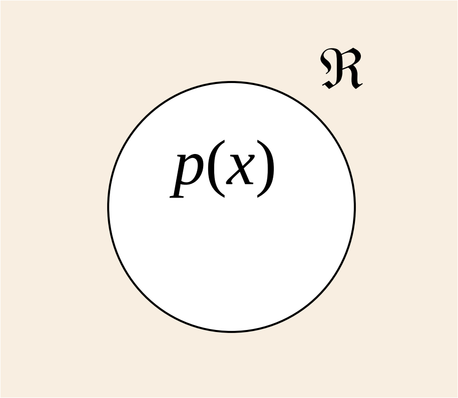
自动化学院

模 式 识 别

概率密度函数的非参数估计

4.1 非参数估计的基本原理

问题：已知样本集 $X=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_N\}$，其中样本均从服从 $p(\boldsymbol{x})$ 的总体中独立抽取 (IID, Independently Identically Distribution)，求 $\hat{p}(\boldsymbol{x})$ 。



考虑随机向量 $\boldsymbol{x}$ 落入区域 $\Re$ 的概率 $P=\int_{\Re} p(x)dx$

N：样本总数　　k：实际落到 $\Re$ 中的样本数　　$\hat{P}=\dfrac{k}{N}$

设 $p(x)$ 在 $\Re$ 内连续，当 $\Re$ 逐渐减小的时候，小到使 $p(x)$ 在其上几乎没有变化时：

$$P=\int_{R} p(x)dx=p(x)V \qquad x\in\Re$$

V：包含 x 的一个小区域 $\Re$ 的体积，有 $V=\int_{R} dx$

$\hat{p}(x)$：为对 $p(x)$ 在小区域内的平均值的估计，即小区域内概率密度估计。

$$\hat{p}(x)=\frac{k}{NV}$$

4.2 直方图法

直方图法（非参数概率密度估计的最简单方法）　$\hat{p}(x)=\dfrac{k}{NV}$

（1）把 x 的每个分量分成 s 个等间隔小窗（若 $x\in E^d$，则形成 s^d 个小舱）

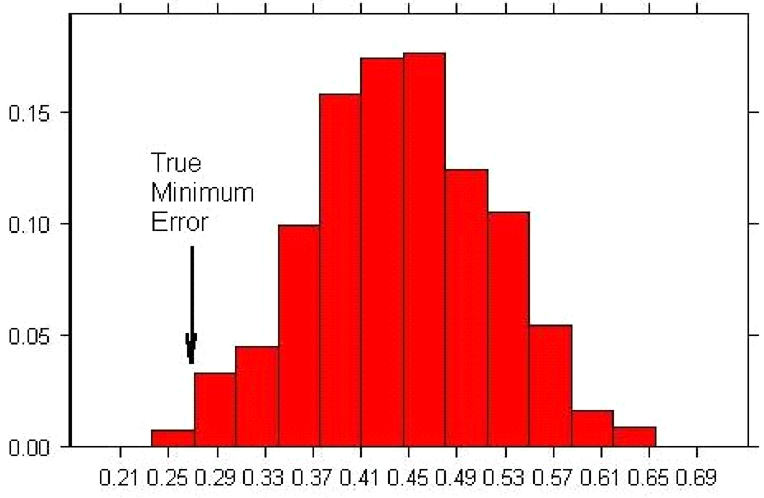
（2）统计落入各个小舱内的样本数 q_i　　$\Re_1,\Re_2,\cdots,\Re_n,\cdots$

（3）相应小舱的概率密度为 $q_i/(NV)$（N：样本总数，V：小舱体积）

V 的选择：过大，估计粗糙；过小，可能某些区域中无样本。

设有样本总数为 $\boldsymbol{n}$，小舱的体积为 V_n，在 $\boldsymbol{x}$ 附近落入小舱的样本个数为 k_n。当 n 趋近于无穷大时，$\hat{p}_n(x)$ 收敛于 $p(x)$ 的条件为：

（1）$\lim\limits_{n\to\infty}V_n=0$；（2）$\lim\limits_{n\to\infty}k_n=\infty$；（3）$\lim\limits_{n\to\infty}\dfrac{k_n}{n}=0$



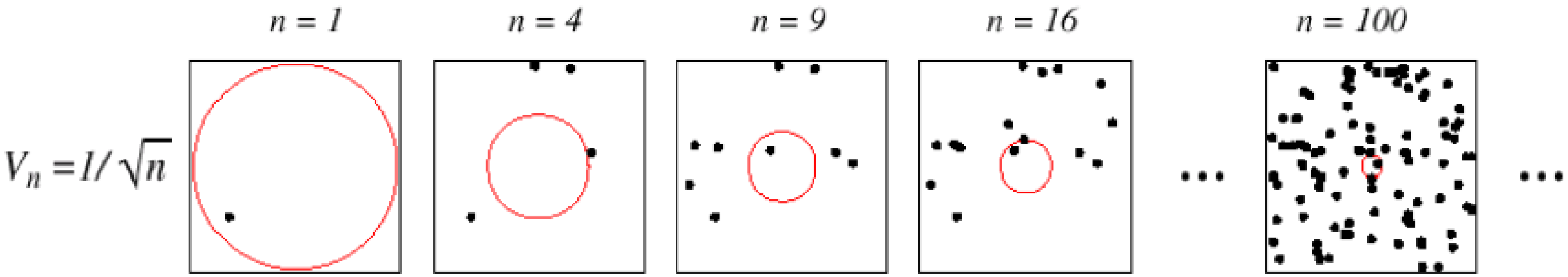
$$\hat{p}_n(x)=\frac{k_n}{nV_n}$$

4.3 Kn近邻估计方法

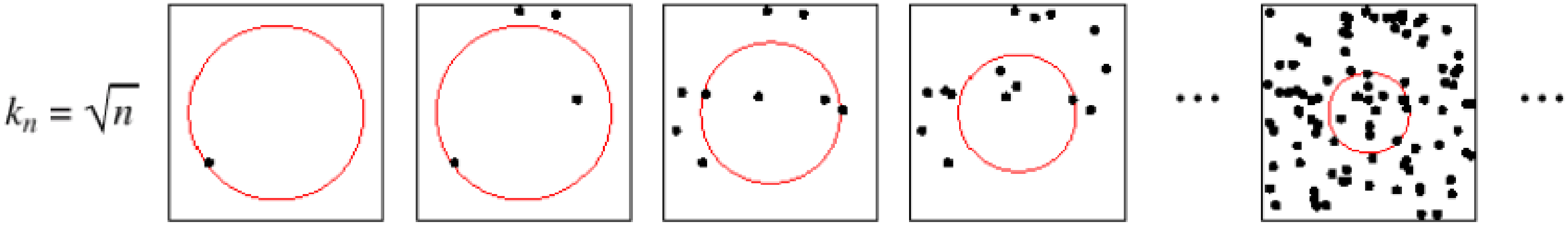
$$\hat{p}_n(x) = \frac{k_n}{nV_n}$$

两种区域选择策略：

（1）选择 V_n （比如 $V_n = \frac{1}{\sqrt{n}}$ ），同时对 k_n 和 $\frac{k_n}{n}$ 加限制以保证收敛— Parzen窗法



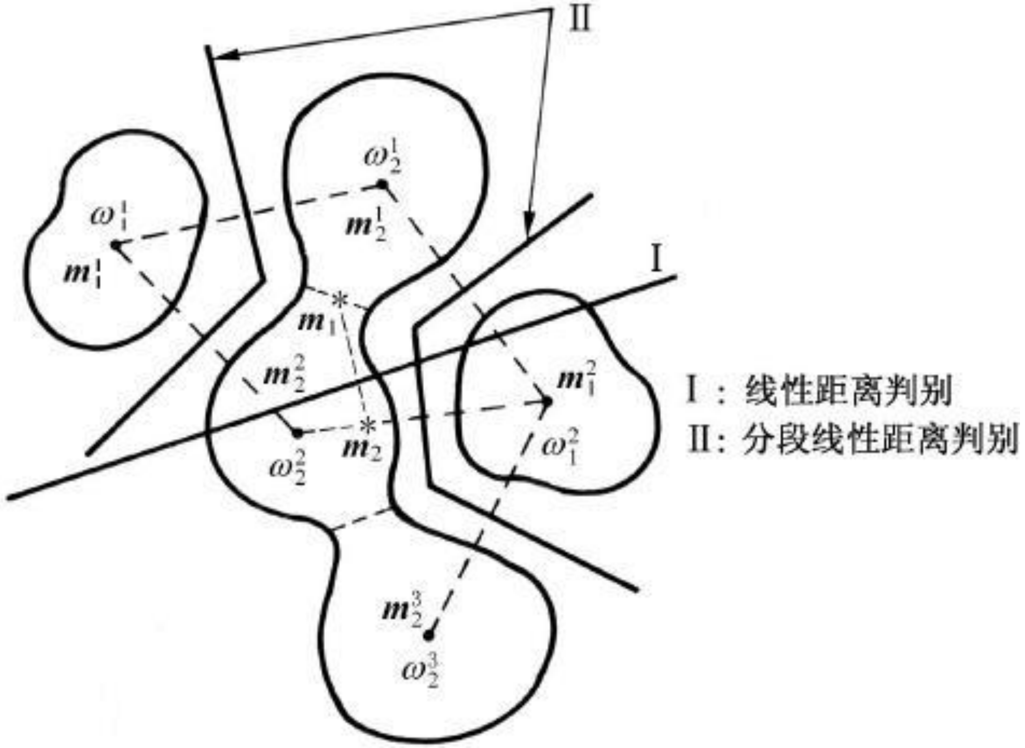
（2）选择 k_n （比如 $k_n = \sqrt{n}$ ）， V_n 为正好包含 x 的 k_n 个近邻— k_N 近邻估计



4.3.1 最近邻法

回顾：最简单的分段线性分类器

把各类划分为若干子类，以子类中心作为类别代表点，考查新样本到各代表点的距离并将它分到最近的代表点所代表的类。



极端情况，将所有样本都作为代表点

最近邻法 ➔ （Nearest-Neighbor method）

4.3.1 最近邻法

样本集 $S_N=\{(x_1,\theta_1),(x_2,\theta_2),\cdots,(x_N,\theta_N)\}$，$x_i$ 为样本，θ_i 为类别标号，且 $\theta_i\in\{1,2,\cdots,c\}$

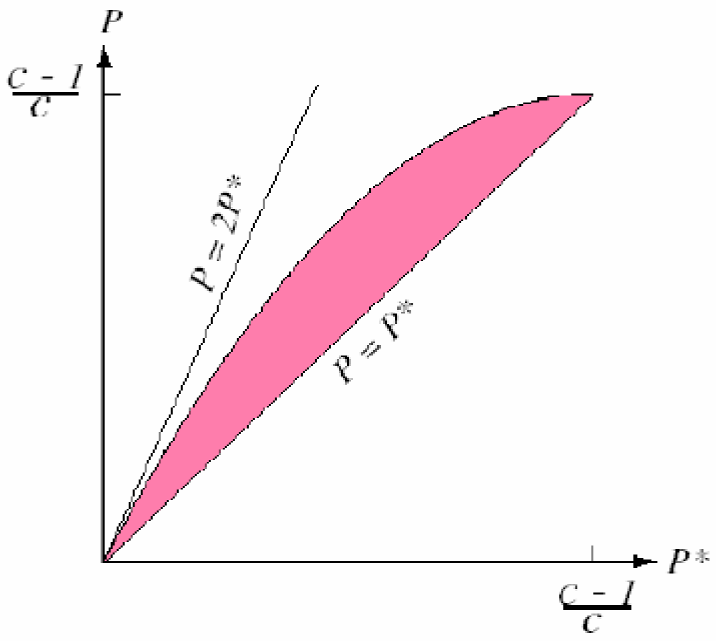
设样本 x_i 与 x_j 之间的距离为 $\delta(x_i,x_j)$（比如欧氏距离 $\|x_i-x_j\|$），对于未知样本 x，设 S_N 中与之距离最近的样本为 x'（类别为 θ'），$\delta(x,x')=\min\limits_{j=1,\cdots,N}\delta(x,x_j)$

此时，将 x 分到 θ' 类，即 $\hat{\omega}(x)=\theta'$。 —— 最近邻决策

4. 3. 1 最近邻法

最近邻法的错误率（渐近分析）

结论： $P^* \le P \le P^*\left(2-\frac{c}{c-1}P^*\right)$



P^*： 贝叶斯错误率

P ： 样本无穷多时最近邻法的错误率（渐近平均错误率）

c ： 类别数

由于一般情况下*P**很小，因此又可粗略表示成： $P^* \le P \le 2P^*$

可粗略说最近邻法的渐近平均错误率在贝叶斯错误率的两倍之内。

- **点x属于θ类，x的最近邻点x’属于θ’类这样两个概率是互相独立的**

$$P\left(\theta,\theta' \mid \mathrm{x},\mathrm{x}'\right)=P\left(\theta \mid \mathrm{x}\right)P\left(\theta' \mid \mathrm{x}'\right)$$

如果我们用最近邻判决规则，我们可能遇到一个错误θ≠θ’，对应的错误率$P_n(e|x,x')$是一个条件概率

$$P_n\left(e \mid \mathrm{x},\mathrm{x}'\right)=1-\sum_{i=1}^{c}P\left(\theta=\omega_i,\theta'=\omega_i \mid \mathrm{x},\mathrm{x}'\right)=1-\sum_{i=1}^{c}P\left(\omega_i \mid \mathrm{x}\right)P\left(\omega_i \mid \mathrm{x}'\right)$$

如果训练样本N趋于无限，空间会被填充满，x等于其最近邻x’的概率为1。

$$\lim_{n\to\infty}P\left(e \mid \mathrm{x},\mathrm{x}'\right)=\lim_{n\to\infty}P\left(e \mid \mathrm{x},\mathrm{x}\right)=\lim_{n\to\infty}P\left(e \mid \mathrm{x}\right)$$

4. 3. 1 最近邻法

$$P=\lim_{n\to\infty}P_n(e)$$

而在这条件下的平均错误率

$$P=\lim_{N\to\infty}P_N(e\,|\,\mathrm{x})$$

$$=\lim_{N\to\infty}\int P_N(e\,|\,\mathrm{x})p(\mathrm{x})dx=\int\lim_{N\to\infty}P_N(e\,|\,\mathrm{x})p(\mathrm{x})dx$$

$$=\int\left[1-\sum_{i=1}^{c}P^2(\omega_i\,|\,\mathrm{x})\right]p(\mathrm{x})dx$$

P称为渐近平均错误率，是P_N(e)在N→∞的极限。

$$\lim_{n\to\infty}P(\omega_i\,|\,\mathrm{x}')\cong\lim_{n\to\infty}P(\omega_i\,|\,\mathrm{x})$$

$$\lim_{n\to\infty}P(e\,|\,\mathrm{x})=\lim_{n\to\infty}P(e\,|\,\mathrm{x},\mathrm{x}')$$

$$P_n(e\,|\,\mathrm{x},\mathrm{x}')=1-\sum_{i=1}^{c}P(\omega_i\,|\,\mathrm{x})P(\omega_i\,|\,\mathrm{x}')$$

贝叶斯错误率的计算式：

$$P^*=\int P^*(e\,|\,\mathrm{x})p(\mathrm{x})dx$$

$$P^*(e\,|\,\mathrm{x})=1-P(\omega_m\,|\,\mathrm{x})$$

$$P(\omega_m\,|\,\mathrm{x})=\max_i\left[P(\omega_i\,|\,\mathrm{x})\right]$$

4. 3. 1 最近邻法

$$\sum_{i=1}^{c} P^2(\omega_i \mid \mathrm{x}) = P^2(\omega_m \mid \mathrm{x}) + \sum_{i \neq m} P^2(\omega_i \mid \mathrm{x})$$

$$\because P(\omega_i \mid \mathrm{x}) \geq 0, \ \sum_{i \neq m} P(\omega_i \mid \mathrm{x}) = 1 - P(\omega_m \mid \mathrm{x}) = P^*(e \mid \mathrm{x})$$

在i≠m时，所有的P都相等，将使得$\sum_{i=1}^{c} P^2(\omega_i \mid \mathrm{x})$最小

$$P(\omega_i \mid \mathrm{x}) = \begin{cases} \dfrac{P^*(e \mid \mathrm{x})}{c-1} & i \neq m \\ 1 - P^*(e \mid \mathrm{x}) & i = m \end{cases}$$

$$\therefore \quad \sum_{i=1}^{c} P^2(\omega_i \mid \mathrm{x}) \geq \left(1 - P^*(e \mid \mathrm{x})\right)^2 + \frac{P^{*2}(e \mid \mathrm{x})}{c-1}$$

$$\Downarrow$$

$$and \quad 1 - \sum_{i=1}^{c} P^2(\omega_i \mid \mathrm{x}) \leq 2P^*(e \mid \mathrm{x}) - \frac{c}{c-1} P^{*2}(e \mid \mathrm{x})$$

4.3.1 最近邻法

$$1-\sum_{i=1}^{c}P^2(\omega_i|\mathrm{x})\le 2P^*(e|\mathrm{x})-\frac{c}{c-1}P^{*2}(e|\mathrm{x})$$

$$\boxed{P=\int\left[1-\sum_{i=1}^{c}P^2(\omega_i|\mathrm{x})\right]p(\mathrm{x})dx}\qquad\boxed{P^*=\int P^*(e|x)p(x)dx}$$

$$P\le 2P^*-\int\frac{c}{c-1}P^{*2}(e|x)p(x)dx$$

$$\int P^{*2}(e|x)p(x)dx=E\left(P^{*2}(e|x)\right)$$

$$E\left(P^{*2}(e|x)\right)=E\left(P^*(e|x)\right)^2+Var\left(P^*(e|x)\right)$$

$$\int P^{*2}(e|x)p(x)dx\le E\left(P^*(e|x)\right)^2=P^{*2}$$

$$P\le P^*\cdot\left(2-\cancel{\frac{c}{c-1}}P^*\right)$$

$$P\le 2P^*$$

4.3.1 最近邻法

$$P_n\left(e\,|\,\mathrm{x},\mathrm{x}'\right)=1-\sum_{i=1}^{c}P\left(\omega_i\,|\,\mathrm{x}\right)P\left(\omega_i\,|\,\mathrm{x}'\right)$$

若是两类问题，则

贝叶斯错误率： $$P^*(e\,|\,X)=\begin{cases}1-P(\omega_1\,|\,X) & if\ \ P(\omega_1\,|\,X)>P(\omega_2\,|\,X)\\ 1-P(\omega_2\,|\,X) & if\ \ P(\omega_1\,|\,X)<P(\omega_2\,|\,X)\end{cases}$$

最近邻法错误率： $$P_N(e\,|\,X)=1-P^2(\omega_1\,|\,X)-P^2(\omega_2\,|\,X)$$

$$\Delta P=P_N(e\,|\,X)-P_N^*(e\,|\,X)$$

$$\Delta P=P(\omega_1\,|\,X)\left[1-P(\omega_1\,|\,X)\right]-P^2(\omega_2\,|\,X)$$

$$=P(\omega_2\,|\,X)\left[P(\omega_1\,|\,X)-P(\omega_2\,|\,X)\right]\qquad if\ \ P(\omega_1\,|\,X)>P(\omega_2\,|\,X)$$

$$\Delta P=P(\omega_2\,|\,X)\left[1-P(\omega_2\,|\,X)\right]-P^2(\omega_1\,|\,X)$$

$$=P(\omega_1\,|\,X)\left[P(\omega_2\,|\,X)-P(\omega_1\,|\,X)\right]\qquad if\ \ P(\omega_1\,|\,X)<P(\omega_2\,|\,X)$$

可见在一般情况下△P是大于零的值。只有在$P(\omega_1|x)=1$或$P(\omega_2|x)=1$ 或$P(\omega_1|x)=P(\omega_2|x)=1/2$ 的情况△P=0.

4. 3. 1 最近邻法

若是两类问题，则

贝叶斯错误率： $P^*(e|x) = \begin{cases} 1-P(\omega_1|x) & if\ P(\omega_1|x) > P(\omega_2|x) \\ 1-P(\omega_2|x) & if\ P(\omega_1|x) < P(\omega_2|x) \end{cases}$

最近邻法错误率： $P_N(e|x) = 1-P^2(\omega_1|x) - P^2(\omega_2|x)$

$$\Delta P = P_N(e|x) - P_N^*(e|x)$$

$$\Delta P = P(\omega_1|X)[1-P(\omega_1|X)] - P^2(\omega_2|X)$$

$$= P(\omega_2|X)[P(\omega_1|X) - P(\omega_2|X)] \qquad if\ P(\omega_1|X) > P(\omega_2|X)$$

$$\Delta P = P(\omega_2|X)[1-P(\omega_2|X)] - P^2(\omega_1|X)$$

$$= P(\omega_1|X)[P(\omega_2|X) - P(\omega_1|X)] \qquad if\ P(\omega_1|X) < P(\omega_2|X)$$

$$P_N(e|x) \geq P^*(e|x)$$

4.3.1 最近邻法

$$1-\sum_{i=1}^{c}P^2(\omega_i\,|\,\mathrm{x})\le 2P^*(e\,|\,\mathrm{x})-\frac{c}{c-1}P^{*2}(e\,|\,\mathrm{x})$$

$$\boxed{P=\int\left[1-\sum_{i=1}^{c}P^2(\omega_i\,|\,\mathrm{x})\right]p(\mathrm{x})d\mathrm{x}}$$

$$P\le P^*\left(2-\frac{c}{c-1}P^*\right)$$

$$P_N(e\,|\,\mathrm{x})\ge P^*(e\,|\,\mathrm{x})$$

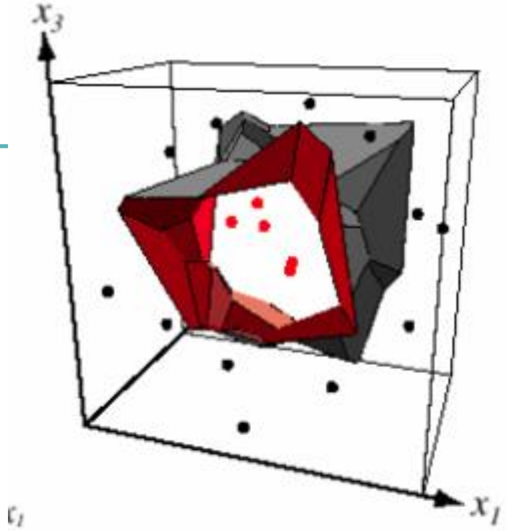
$$P^*\le P\le P^*(2-\frac{C}{C-1}P^*)$$

$$P^*\le P\le 2P^*$$

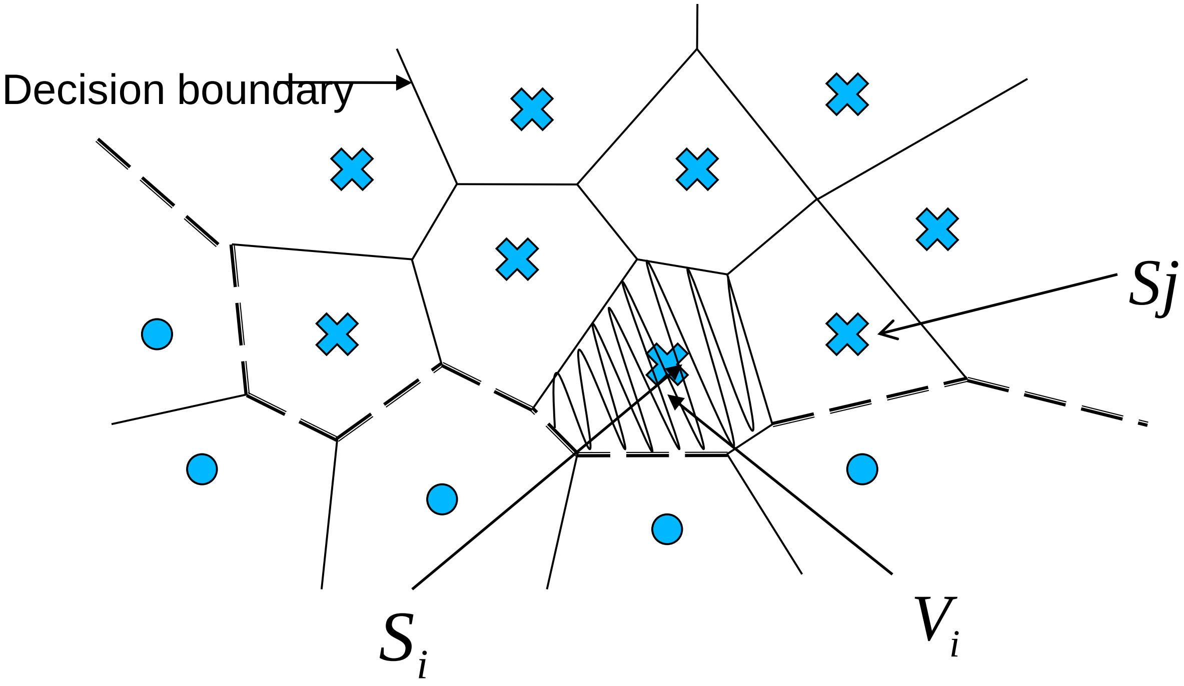
P^*：贝叶斯错误率　P：最近邻法错误率　C：类别数

4.3.1 最近邻法

Voronoi图：又名泰森多边形，是由一组连接两邻近点直线的垂直平分线组成的连续多边形组成。



Voronoi 网格：最近邻规则基于训练样本把特征空间分成一个个网格单元结构，称为Voronoi网格。



- V_i 是一个多边形,任何落入这个多边形的点距离点S_i都比其它已知样本点的距离更近。
- 每个网格包含一个训练样本点，该网格中任何一个位置距离该训练样本点都比距离其它网格的训练样本点更近。
- 如果测试样本 x 落入该网格，则判别为该网格样本点S_i所属的类别。

4.3.2 k-近邻法（kNN）

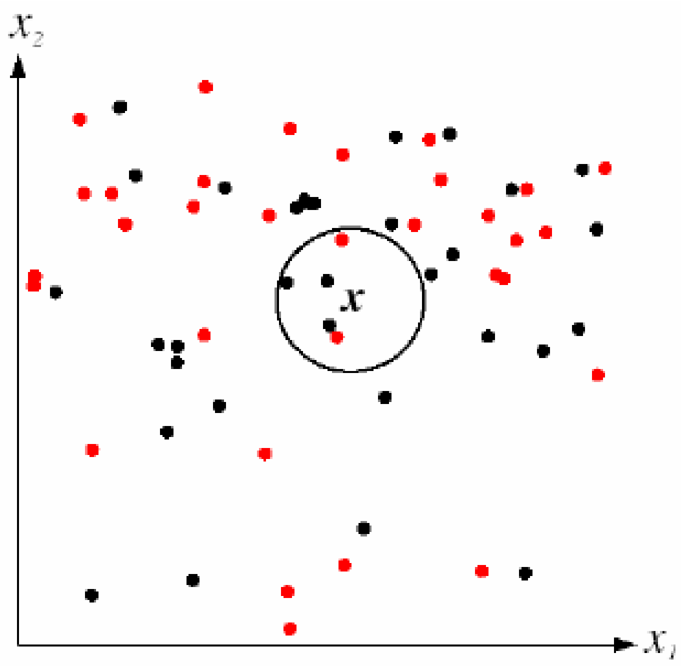
最近邻法的推广

k-近邻法： 找出 **x** 的 **k** 个近邻，看其中多数属于哪一类，则把 **x** 分到哪一类。

N 个样本，包含 **c** 个类别 $\omega_i \ (i=1,\cdots,c)$， k_i 为 **x** 的 **k** 个近邻中属于 ω_i 的样本数。

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

决策规则： **If** $g_j(x) = \max\limits_{i=1,\cdots,c} k_i$ ， **then** $x \in \omega_j$



x 的分类，是通过统计最邻近的 **k** 个样本的属性，用投票法将最常见的类别标示 **x**。

4.3.2 k-近邻法（kNN）

渐近平均错误率的界：

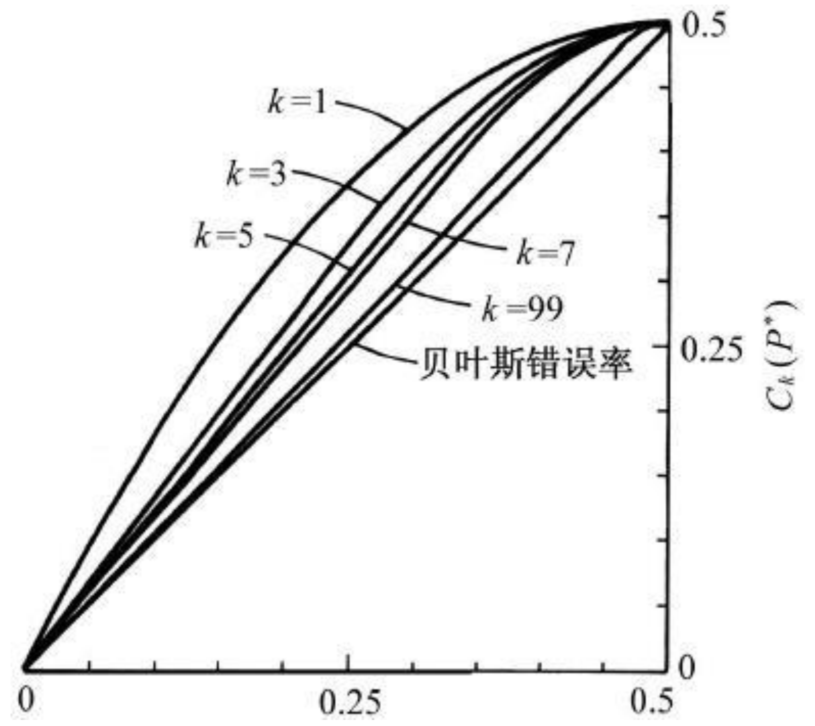
N 无穷大时，k 越大,P_k 的上限越低（越靠近下限）。但 k 应始终是 N 中的一小部分，保证 k个近邻均充分接近 x 。否则这一关系不成立。

一般来说，总有

$$P^* \le P_k \le P^*\left(2 - \frac{c}{c-1}P^*\right)$$

或者简化为

$$P^* \le P_k \le 2P^*$$



4.3.2 k-近邻法（kNN）

✓ **如果k近邻大多数被标记为ω_m类，作出这样选择的概率为（两类问题）**

$$\sum_{i=(k+1)/2}^{k}\binom{k}{i}P\left(\omega_m|\mathrm{x}\right)^{i}\left[1-P\left(\omega_m|\mathrm{x}\right)\right]^{k-i}$$

可以证明，如果k是奇数，大样本的两类问题的k近邻规则错误率是有界的,它可以用函数$C_k(P^*)$表示, 这里$C_k(P^*)$被定义为关于P *的最小的凹函数

4.3.2 k-近邻法（kNN）

最近邻法条件错误率（两类问题）

$$P_n\left(e|\mathrm{x},\mathrm{x}'\right)=1-\sum_{i=1}^{c}P\left(\theta=\omega_i,\theta'=\omega_i|\mathrm{x},\mathrm{x}'\right)=1-\sum_{i=1}^{c}P\left(\omega_i|\mathrm{x}\right)P\left(\omega_i|\mathrm{x}'\right)$$

$$=1-P\left(\omega_1|\mathrm{x}\right)P\left(\omega_1|\mathrm{x}'\right)-P\left(\omega_2|\mathrm{x}\right)P\left(\omega_2|\mathrm{x}'\right)$$

$$=1-\left(1-P\left(\omega_2|\mathrm{x}\right)\right)P\left(\omega_1|\mathrm{x}'\right)-\left(1-P\left(\omega_1|\mathrm{x}\right)\right)P\left(\omega_2|\mathrm{x}'\right)$$

$$=\cancel{1}-\cancel{P}\left(\omega_1|\mathrm{x}'\right)+P\left(\omega_2|\mathrm{x}\right)P\left(\omega_1|\mathrm{x}'\right)\cancel{-P}\left(\omega_2|\mathrm{x}'\right)+P\left(\omega_1|\mathrm{x}\right)P\left(\omega_2|\mathrm{x}'\right)$$

$$=P\left(\omega_1|\mathrm{x}\right)P\left(\omega_2|\mathrm{x}'\right)+P\left(\omega_2|\mathrm{x}\right)P\left(\omega_1|\mathrm{x}'\right)$$

$\because N\to\infty$时，有$P\left(\omega_i|x'\right)\doteq P\left(\omega_i|x\right)$

$$\therefore P_{n\to\infty}\left(e|\mathrm{x},\mathrm{x}'\right)=P\left(\omega_1|\mathrm{x}\right)P\left(\omega_2|\mathrm{x}\right)+P\left(\omega_2|\mathrm{x}\right)P\left(\omega_1|\mathrm{x}\right)$$

4.3.2 k-近邻法（kNN）

推广到k近邻， 设x属于ω_1,但$k_1 \le \frac{k-1}{2}$,则$k_1 \le k-k_1 = k_2$,此时发生误判，

这一事件的概率为$\sum_{i=0}^{(k-1)/2} \binom{k}{j} P(\omega_1 | \mathrm{x})^j P(\omega_2 | \mathrm{x})^{k-j}$

反之， x属于ω_2,发生误判的概率为$\sum_{i=0}^{(k-1)/2} \binom{k}{j} P(\omega_2 | \mathrm{x})^j P(\omega_1 | \mathrm{x})^{k-j}$

所以有给定x时的条件错误率为

$$P_{k,N}(e|x) = P(\omega_1 | x) \sum_{i=0}^{(k-1)/2} \binom{k}{j} P(\omega_1 | x)^j P(\omega_2 | x)^{k-j}$$

$$+ P(\omega_2 | x) \sum_{i=0}^{(k-1)/2} \binom{k}{j} P(\omega_2 | x)^j P(\omega_1 | x)^{k-j}$$

4.3.2 k-近邻法（kNN）

$$P_{k,N}(e|x)=P(\omega_1|x)\boxed{\sum_{j=0}^{(k-1)/2}\binom{k}{j}P(\omega_1|x)^j P(\omega_2|x)^{k-j}}$$

$$+P(\omega_2|x)\boxed{\sum_{j=0}^{(k-1)/2}\binom{k}{j}P(\omega_2|x)^j P(\omega_1|x)^{k-j}}$$

$x\in\omega_1$而决策为 $x\notin\omega_1$的概率

$x\in\omega_2$而决策为 $x\notin\omega_2$的概率

一般化

$$P_{k,N\to\infty}(e|x)=P(\omega_i|x)\sum_{j=0}^{(k-1)/2}\binom{k}{j}P(\omega_i|x)^j\left[1-P(\omega_i|x)\right]^{k-j}$$

$$+\underline{\left[1-P(\omega_i|x)\right]\sum_{j=(k-1)/2}^{k}}\binom{k}{j}P(\omega_i|x)^j P\left[1-P(\omega_i|x)\right]^{k-j}$$

一般化

$$P_{k,N\to\infty}(e|x)=P(\omega_i|x)\sum_{j=0}^{(k-1)/2}\binom{k}{j}P(\omega_i|x)^j\left[1-P(\omega_i|x)\right]^{k-j}$$
$$+\left[1-P(\omega_i|x)\right]\sum_{j=(k-1)/2}^{k}\binom{k}{j}P(\omega_i|x)^j\left[1-P(\omega_i|x)\right]^{k-j}$$

贝叶斯条件错误率为

$$P^*(e|x)=\min\left[P(\omega_1|x),P(\omega_2|x)\right]=\min\left[P(\omega_1|x),1-P(\omega_1|x)\right]$$

组合$\binom{k}{j}=\frac{k!}{j!(k-j)!}=$组合$\binom{k}{k-j}$

$$P_{k,N\to\infty}(e|x)=P^*(e|x)\sum_{j=0}^{(k-1)/2}\binom{k}{j}P^*(e|x)^j\left[1-P^*(e|x)\right]^{k-j}$$
$$+\left[1-P^*(e|x)\right]\sum_{j=0}^{(k-1)/2}\binom{k}{j}P^*(e|x)^{k-j}\left[1-P^*(e|x)\right]^j$$
$$=\boxed{\sum_{i=0}^{(k-1)/2}\binom{k}{j}\left[\left(P^*\right)^{j+1}\left(1-P^*\right)^{k-j}+\left(P^*\right)^{k-j}\left(1-P^*\right)^{j+1}\right]}$$

4.3.2 k-近邻法（kNN）

- 定义一个贝叶斯条件错误率P*(e|x)的函数C_k[P*(e|x)]，C_k[P*(e|x)]为大于 $P_{k,N\to\infty}(e|x)$ 的最小凹函数，那么对于所有x有

K近邻渐进平均错误率

$$P_{k,N\to\infty}(e|\mathrm{x})\le C_k\left[P^*(e|\mathrm{x})\right]$$

- 因为 $P_{k,N\to\infty}(e|x)$ 随k的增大单调减小，故最小凹函数C_k也随k单调减小，所以有

$$\therefore 有 P=E[P_{k,N\to\infty}(e|\mathrm{x})]\le E\left\{C_k\left[P^*(e|\mathrm{x})\right]\right\}\le C_k\left\{E\left[P^*(e|\mathrm{x})\right]\right\}=C_k\left[P^*\right]$$

$$P^*\le P\le C_k\left[P^*\right]\le C_{k-1}\left[P^*\right]\le\cdots\le C_1\left[P^*\right]\le 2P^*\left(1-P^*\right)$$

贝叶斯错误率

最近邻法和k-近邻法的错误率上下界都是在一倍到两倍贝叶斯决策方法的错误率范围内。

在k >1的条件下，k-近邻法的错误率要低于最近邻法。

得证

在$k\to\infty$的条件下，k-近邻法的错误率等于贝叶斯误差率

4.3.2 k-近邻法（kNN）

已知二维模式识别样本有

Prototypes	Labels
$(1.5, 2.8)^T$	ω_1
$(1.0, 2.4)^T$	ω_2
$(1.5, 2.5)^T$	ω_5
$(2.0, 2.8)^T$	ω_2
$(1.3, 2.6)^T$	ω_5
$(1.8, 2.0)^T$	ω_1
$(1.2, 2.8)^T$	ω_5
$(1.2, 2.2)^T$	ω_1

待识别样本 $x = (1.2, 2.5)^T$，现按照欧氏距离按 k-近邻规则进行模式分类（k=5）。

4.3.2 k-近邻法（kNN）

已知二维模式识别样本有

Prototypes	Labels
$(1.5, 2.8)^T$	ω_1
$(1.0, 2.4)^T$	ω_2
$(1.5, 2.5)^T$	ω_5
$(2.0, 2.8)^T$	ω_2
$(1.3, 2.6)^T$	ω_5
$(1.8, 2.0)^T$	ω_1
$(1.2, 2.8)^T$	ω_5
$(1.2, 2.2)^T$	ω_1

解：按照欧氏距离判断，待选点包括：

Prototypes	Labels
$(1.0, 2.4)^T$	ω_2
$(1.5, 2.5)^T$	ω_5
$(1.3, 2.6)^T$	ω_5
$(1.2, 2.8)^T$	ω_5
$(1.2, 2.2)^T$	ω_1

则待识别样本属于类别 5。

待识别样本 $x = (1.2, 2.5)^T$，现按照欧氏距离按 k-近邻规则进行模式分类（k=5）。

4.3.3 近邻法的快速算法

问题：

① 存储量和计算量

② 票数接近时风险较大，有噪声时风险加大

③ 有限样本下性能如何？

改进：

① 减少计算量和存储量

② 引入拒绝机制

③ 根据实际问题修正投票方式，如加权投票，否决票等

可考虑距离加权，考虑样本比例及先验概率等

4.3.3 近邻法的快速算法

近邻法在计算上的问题：
- 需存储所有训练样本
- 新样本需与每个样本做比较

加速方法:

- **“部分距离”计算法**
- **“预建立结构 ”算法**

4.3.3 近邻法的快速算法

“部分距离” 计算法

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

$$r \le d$$

- 一旦其部分的距离大于目前最接近的样本的全欧氏距离 $(r = d)$时，终止距离计算。

4.3.3 近邻法的快速算法

近邻法在计算上的问题：
- 需存储所有训练样本
- 新样本需与每个样本做比较

加速方法:

- **“部分距离”计算法**
- **“预建立结构 ”算法**

改进的思路：

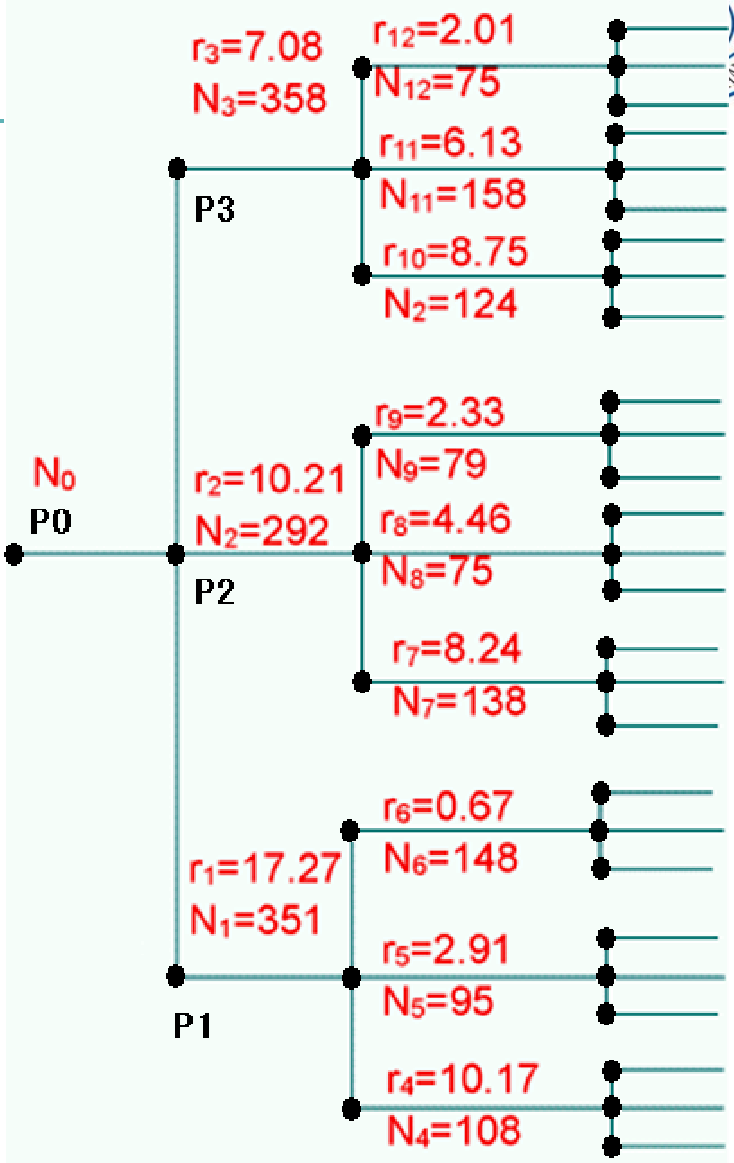
- 对样本集进行组织与整理，分群分层，尽可能将计算压缩到在接近测试样本邻域的小范围内，避免盲目地与训练样本集中每个样本进行距离计算。
- 在原有样本集中挑选出对分类计算有效的样本，使样本总数合理地减少，以同时达到既减少计算量，又减少存储量的双重效果。

4.3.3 近邻法的快速算法

“预建立结构”计算法

（1）样本集的分级分解构建搜索树

- 将整个样本集分成 l 个子集，每个子集又分为它的 l 个子集，如此进行若干次就能建立起一个样本集的树形结构。
- 分成子集的原则是该子集内的样本尽可能聚成堆，这可用聚类方法实现。
- 计算并存储 $\mathcal{X}_p$ 的 M_p、r_p 及 $D(x_i, M_p)$



P: 树的一个结点，对应一个样本子集χ_p

$N_p : \chi_p$中的样本数

$M_p : \chi_p$中的样本均值

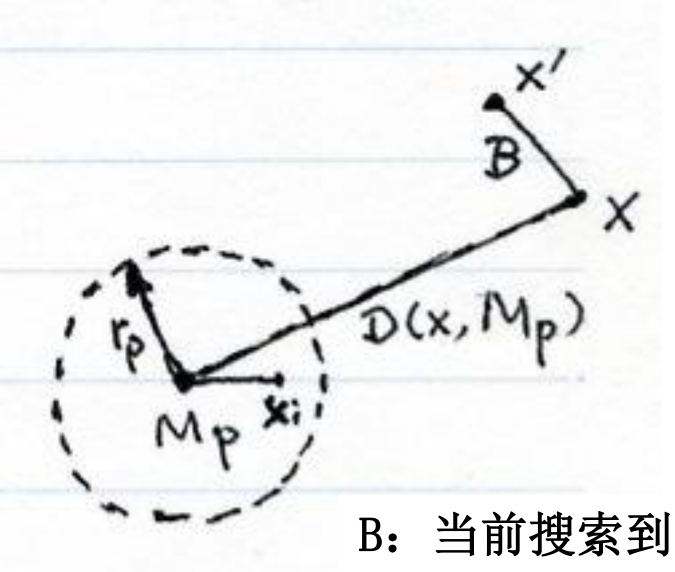
r_p： 从χ_p中任一样本到M_p的最大距离 $r_p = \max\limits_{x_i \in \mathcal{X}_p} D(x_i, M_p)$

4.3.3 近邻法的快速算法

（2）搜索（分支定界算法）

搜索规则：

1. 对新样本 x，结点 $\mathcal{X}_p$，若 $D(x,M_p)>B+r_p$
   则 x 的近邻不可能在 $\mathcal{X}_p$ 中
2. 对新样本 x，结点 $\mathcal{X}_p$ 中的样本 $x_i\in\mathcal{X}_p$
   若 $D(x,M_p)>B+D(x_i,M_p)$，则 x_i 不是 x 的最近邻。



B：当前搜索到的最近邻距离

其中 r_p, $D(x_i,M_p)$ 在训练（建树）过程中可以先计算保存，搜索过程只需计算 $D(x,M_p)$ 或更新 B。

- 这种方法着眼于只解决减少计算量，但没有达到减少存储量的要求。
- 如果结构合理，可以降低计算时间。

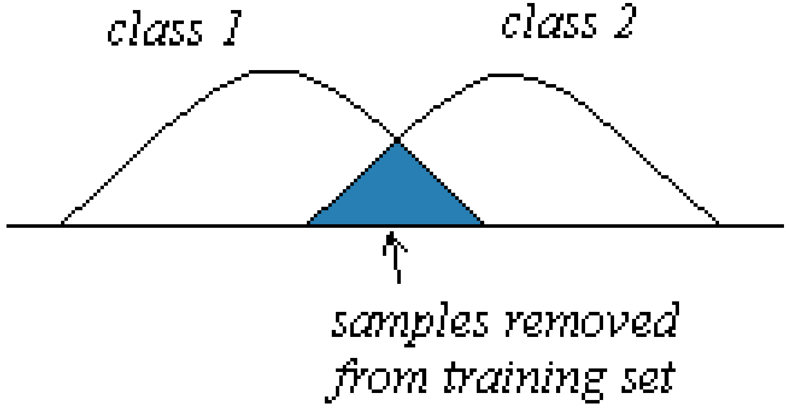
4.3.4 剪辑近邻法

基本理解：

获得更准确的错误率

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。应将其从样本集中去掉。

- 考查样本是否为可能的误导样本
- 考查方法是通过试分类，认为错分样本为误导样本
- 若是则从样本集中去掉——剪辑



4.3.4 剪辑近邻法

基本做法：

将样本集分为考试集 X^{NT} 和参考集 X^{NR}： $X^{N}=X^{NT}\cup X^{NR}$ ， $X^{NT}\cap X^{NR}=\phi$

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} 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。

- **训练样本和测试样本没有独立性，会产生一个偏于乐观的估计。**

4.3.4 剪辑近邻法

错误率分析（渐近错误率）

1. 若用最近邻剪辑，用最近邻分类，则错误率

$$P_1^E(e|x) = \frac{P(e|x)}{2[1-P(e|x)]} \qquad \because P(e|x) \ll 0.5$$

即 $P_1^E(e) \le P(e)$ （$P(e|x)$、$P(e)$是没有剪辑的近邻法的错误率）

当$P(e)$很小时，如 $P(e) < 0.1$，则有 $P_1^E(e) \approx \frac{1}{2}P(e)$

而 $P(e) \le 2P^*$（ P^*为贝叶斯错误率）。

故此时 $P_1^E(e)$ 接近 P^*。

4.3.4 剪辑近邻法

2. 若用 k 近邻剪辑，用最近邻分类，则

$$P_k^E(e|x)=\frac{P(e|x)}{2[1-P_k(e|x)]}<P_1^E(e|x)$$

当 $k\to\infty$ 时 $P_k^E(e)$ 收敛于 P^*（N应更快地趋向∞ ）

3. 多类情况，多类剪辑近邻错误率 $P_{k_c}^E(e|x)$ 小于两类情况

4. 重复剪辑

样本足够多时，可多次重复剪辑，效果更好。

4.3.4 剪辑近邻法

一种重复剪辑算法——MULTIEDIT：

（1）（散开）把 X^N随机划分为 s个子集，$X_1,\cdots,X_s$，$s \geq 3$

（2）（分类）用 X_j（$j=(i+1)mod(s)$）对 X_i 中的样本分类

$i=1,\cdots,s$．

（3）（剪辑）去掉(2)中错分的样本

（4）（混合）将剩下的样本合在一起，形成新的 $X^N(X^{NE})$

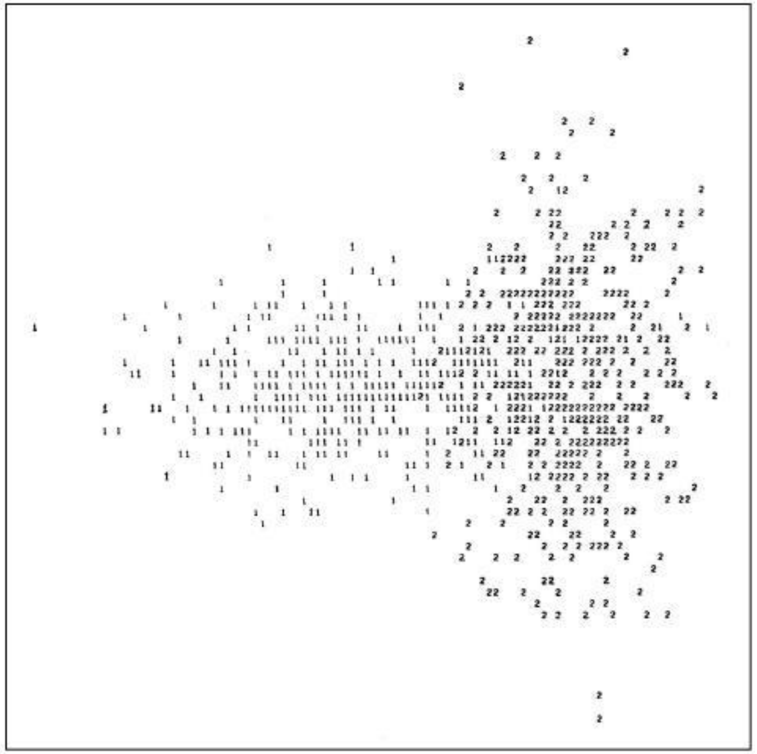
（5）（终止）如果该次迭代都没有样本被剪掉，则停止；否则用新的 X^N转（1）。

算法停止后，用最后的 X^{NE} 作为分类的样本集。

由此可见，每次迭代过程都要重新对现有的样本集进行重新随机划分，保证了剪辑的独立性。

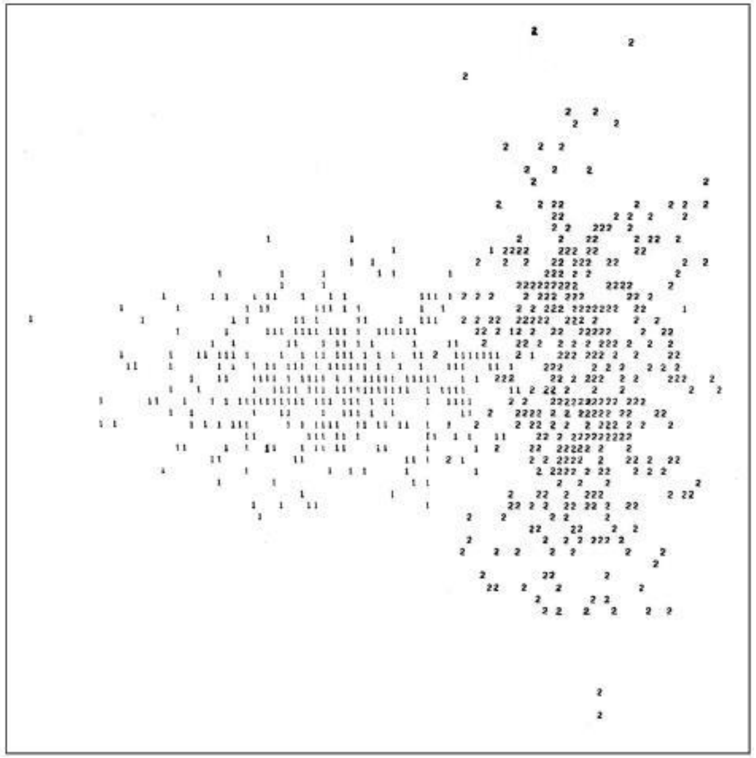
4.3.4 剪辑近邻法

例：



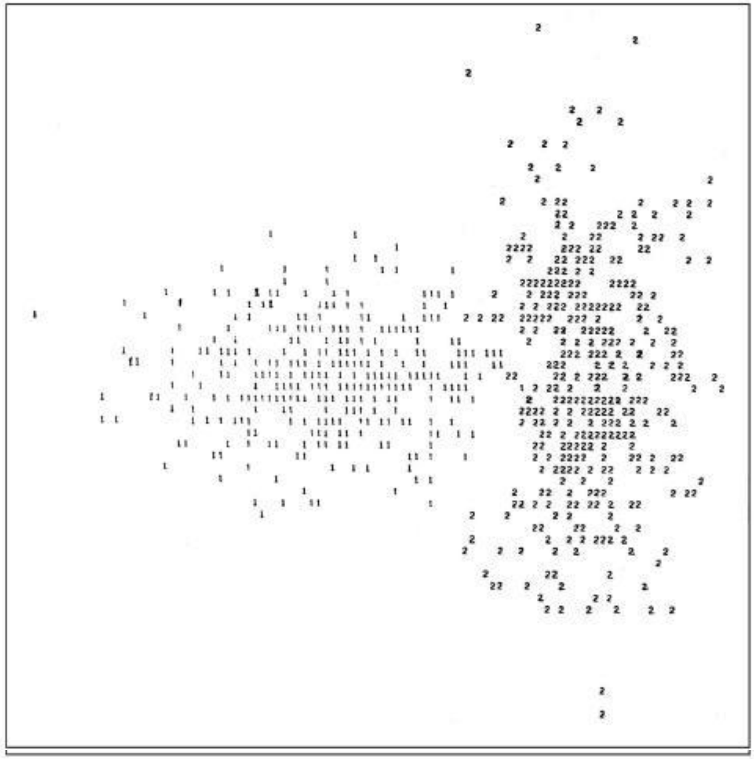
4.3.4 剪辑近邻法

例：



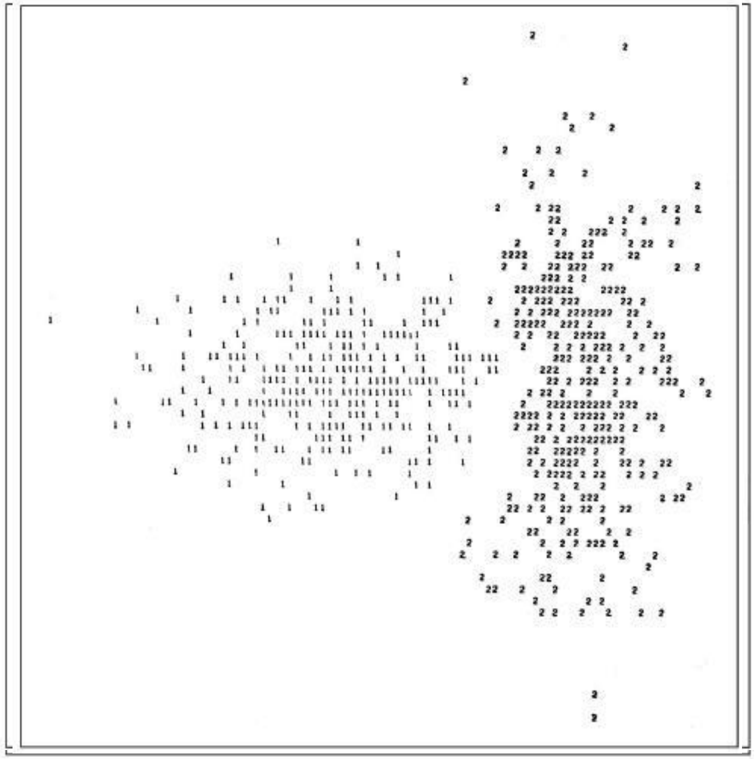
4.3.4 剪辑近邻法

例：



4.6.4 剪辑近邻法

例：



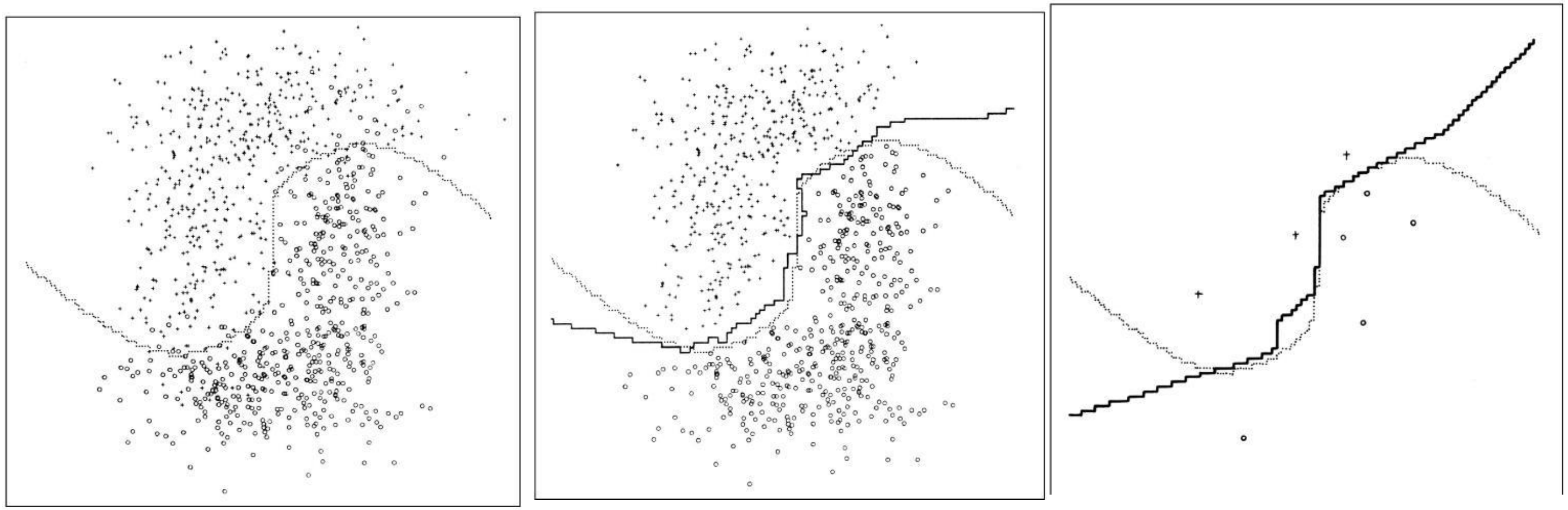
4.3.5 压缩近邻法

主要用以减少计算量

将 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作分类的样本集。

可与剪辑法配合使用。

例：



4.3.5 压缩近邻法

```
% =====================压缩剪辑近邻算法（Condensing）=====================
% s:  划分的子集数目
% Xn: 当前样本集
% Xcur: 当前样本集经一次迭代后的样本集
% Xi: 当前考试集
% Xr: 当前参考集
% K:  退出控制条件，迭代K次，若没有样本被剪辑掉，则退出
% ======================================================================
clear,close all;
X = [randn(300,2)+ones(300,2);...
     randn(300,2)-ones(300,2);];
X(1:300,3)=1;X(301:600,3)=2;
% ======================================================================
figure, plot(X(1:300,1),X(1:300,2),'r.')
hold on,plot(X(301:600,1),X(301:600,2),'b.')
title('初始样本分布图')
% ======================================================================
s=3;Xcur=X;loop=0;Xold=X;K=5;
```

4.3.5 压缩近邻法

```
% ==
% s:
% Xn
% Xc
% Xi
% Xr
% K:
% ==
clea
X =

X(1:
% ==
figu
hold
titl
% ==
s=3;
```

```
while loop<K
    Xn=Xcur;
    Xold=Xcur;
    Xcur=[];
    [row1,col]=size(Xn);
    uu=unifrnd(0,s,row1,1);%产生row1行1列的随机数，随机数的范围在0-s之间
    uu=ceil(uu);%取整，方向是使数据变大
    for i=1:s    %样本随机划分为s个子集
        Xi=Xn((uu==i),:);%test set %Xi为考试集
        r=mod(i+1,s);%取余数
        if r==0
            r=s;
        end
        Xr=Xn((uu==r),:);%reference set%Xr为训练集
        [row,col]=size(Xi);
        j=1;
        while j<=row
            [rClass,jClass]=NNforCondense(Xr,Xi(j,:));%用训练集中的样本对考试集中的样本进行最近邻分类
            if rClass~=jClass%如果类别不同，则从考试集中分类错误的样本去除
                Xi(j,:)=[];
                row=row-1;
            else
                j=j+1;
            end
        end
```

4. 3. 5 压缩近邻法

```
        Xcur=[Xcur;Xi];
    end
    [oldRow,col]=size(Xold);
    [curRow,col]=size(Xcur);
    if oldRow==curRow
        loop=loop+1;
    else
        loop=0;
    end
end
% ================================================================
%把当前样本集Xcur中的元素按原类别分类
[row,col]=size(Xcur);
Xcur1=[];Xcur2=[];
tic
for i=1:row
    if Xcur(i,3)==1
        Xcur1=[Xcur1;Xcur(i,1:2)];
    elseif Xcur(i,3)==2
        Xcur2=[Xcur2;Xcur(i,1:2)];
    end
end
time1=toc;
figure, plot(Xcur1(:,1),Xcur1(:,2),'r.')
hold on,plot(Xcur2(:,1),Xcur2(:,2),'b.')
title('剪辑后样本分布图')
```

4.3.5 压缩近邻法

```
            Xcur
        end
        [oldRow,
        [curRow,
        if oldR
            loop
        else
            loop
        end
end
% =========
%把当前样本集
[row,col]=s
Xcur1=[];Xcu
tic
for i=1:row
        if Xcur
            Xcur
        elseif X
            Xcur
        end
end
time1=toc;
figure, plot
hold on,plot
title('剪辑
```

```
% ===================Condensing===================================
Xstore=Xcur(1,:);
Xgab=Xcur(2:row,:);
while 1
    Xoldstore=Xstore;
    [row,col]=size(Xgab);
    j=1;
    while j<=row
        [sClass,gClass]=NNforCondense(Xstore,Xgab(j,:));
        if sClass~=gClass
            Xstore=[Xstore;Xgab(j,:)];
            Xgab(j,:)=[];
            row=row-1;
        else
            j=j+1;
        end
    end
    [oldRow,col]=size(Xoldstore);
    [curRow,col]=size(Xstore);
    [gRow,rCol]=size(Xgab);
    if oldRow==curRow | gRow*rCol==0
        break;
    end
end
```

4.3.5 压缩近邻法

```
        Xcur
    end
    [oldRow,
    [curRow,
    if oldR
        loop
    else
        loop
    end
end
% ==========
%把当前样本集
[row,col]=s
Xcur1=[];Xcu
tic
for i=1:row
    if Xcur
        Xcur
    elseif X
        Xcur
    end
end
time1=toc;
figure, plot
hold on,plot
title('剪辑后
```

```
% =======
Xstore=Xc
Xgab=Xcur
while 1
    Xolds
    [row,
    j=1;
    while
        [
        i
        e
        end
    end
    [oldRow,col]=size(Xoldstore);
    [curRow,col]=size(Xstore);
    [gRow,rCol]=size(Xgab);
    if oldRow==curRow | gRow*rCol==0
        break;
    end
end
```

```
Xcurstore1=[];Xcurstore2=[];
[curRow,col]=size(Xstore);
for i=1:curRow
    if Xstore(i,3)==1
        Xcurstore1=[Xcurstore1;Xstore(i,1:2)];
    else
        Xcurstore2=[Xcurstore2;Xstore(i,1:2)];
    end
end
figure, plot(Xcurstore1(:,1),Xcurstore1(:,2),'r.')
hold on,plot(Xcurstore2(:,1),Xcurstore2(:,2),'b.')
axis([-4 5,-4 5]);
title('压缩后样本分布图')
```

4.3.5 压缩近邻法

```
        Xcu
    end
    [oldRow,
    [curRow,
    if oldR
        loo
    else
        loo
    end
end
% =========
%把当前样本集
[row,col]=s
Xcur1=[];Xcu
tic
for i=1:row
    if Xcur
        Xcu
    elseif X
        Xcu
    end
end
time1=toc;
figure, plot
hold on,plot
title('剪辑
```

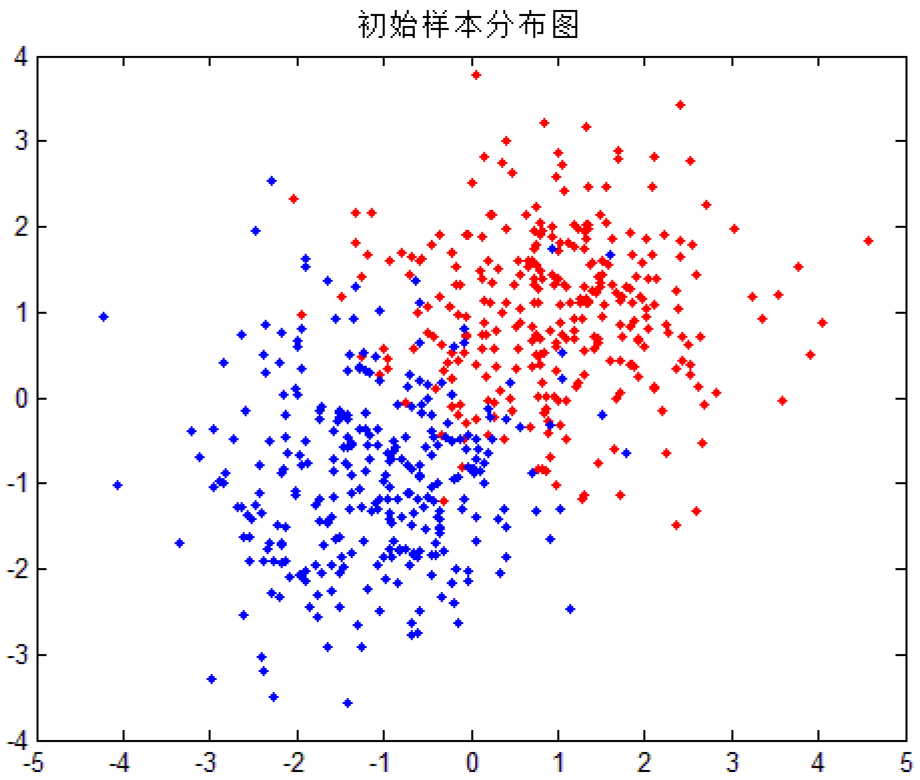
```
% ======
Xstore=Xc
Xgab=Xcur
while 1
    Xolds
    [row,
    j=1;
    while
        [
        i
        e
    end
    end
    [oldRow,col]=s
    [curRow,col]=s
    [gRow,rCol]=si
    if oldRow==cur
        break;
    end
end
```

```
Xcurs
[curR
for i
    i
    e
    e
end
figur
hold
axis(
title
```

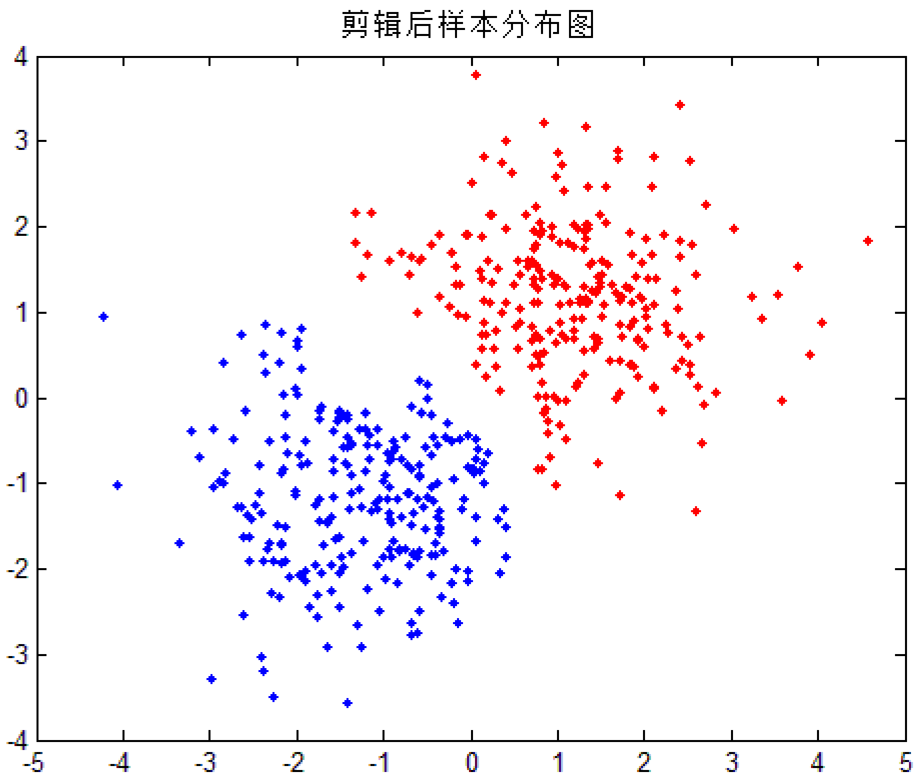
```
% ----一般近邻算法---
% num:     每类的样本数目
% rClass: 返回值，x在Xr中最近邻的样本类别
% xClass: 返回值，x的样本类别
% ==========================================================
function [rClass,xClass]=NNforCondense(Xr,x)

tic
% X = [randn(200,2)+ones(200,2);...
%      randn(200,2)-2*ones(200,2);...
%      randn(200,2)+4*ones(200,2);];
%  x=randn(1,2);%待判样本
[row,col]=size(Xr);
Xdist=zeros(row,1);
for i=1:row
    Xdist(i)=norm(x(1,1:2)-Xr(i,1:2))^2;
end
[Xdist,ind]=sort(Xdist,'ascend');
B=dist(1);
Xnn=Xr(ind(1),:);
rClass=Xnn(1,3);
xClass=x(1,3);
times=toc;
end
```

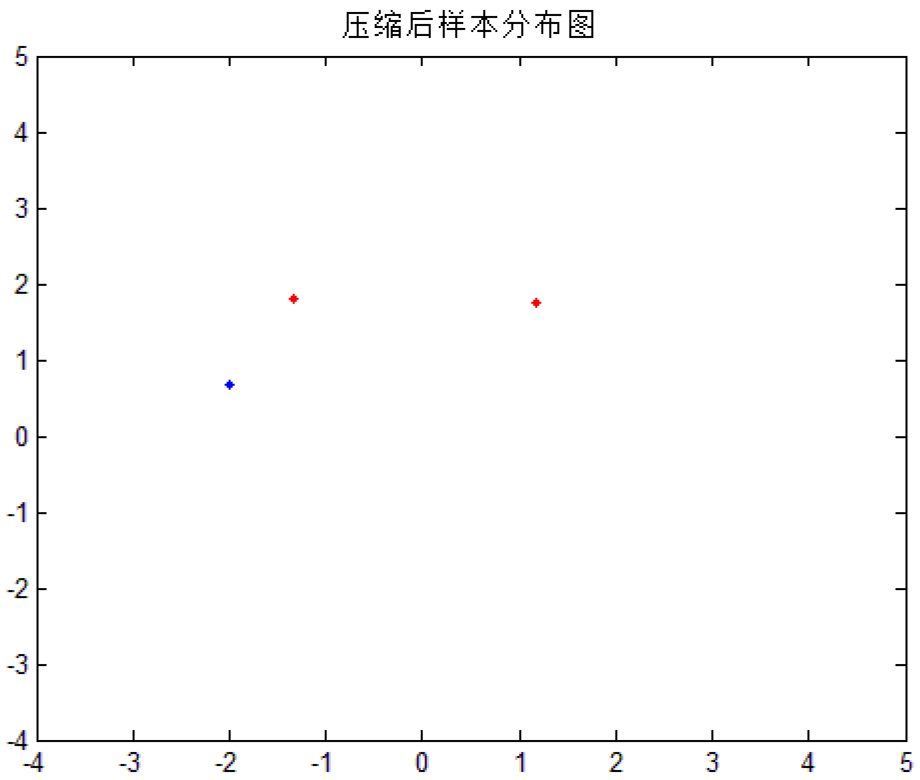
4.3.5 压缩近邻法



4.3.5 压缩近邻法



4.3.5 压缩近邻法



4.3.6 可做拒绝决策的近邻法

由于近邻法决策实际只取决于个别样本，因此有时风险较大，尤其是最近邻法和 k 近邻法当两类近邻数接近时，为此，可考虑引入拒绝决策。

方法： 设某个 $k' > \frac{1}{2}(k+1) \quad (k' < k)$ ，

只有当 x 的 k 个近邻中有大于或等于 k' 个属于 ω_i 类时，才决策 $x \in \omega_i$ ，否则拒绝.

—— 简单多数 ⇒绝对多数

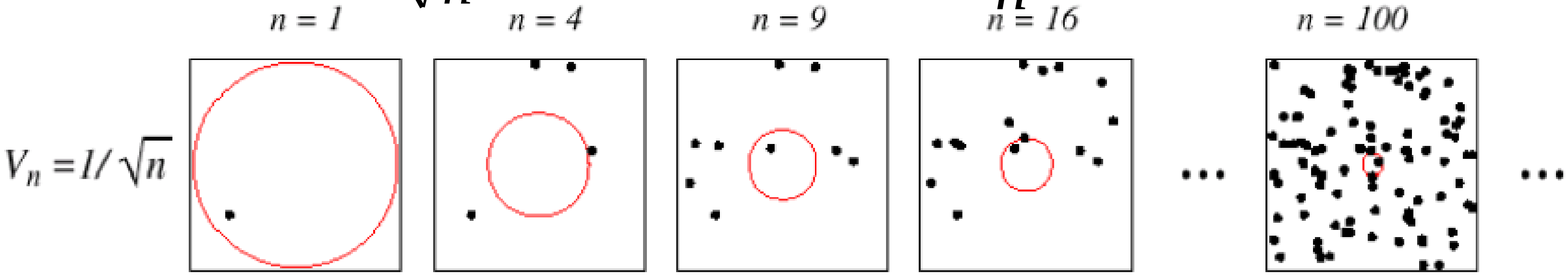
拒绝决策同样可引入改进的近邻法中，比如剪辑近邻法。

4.4 Parzen窗法

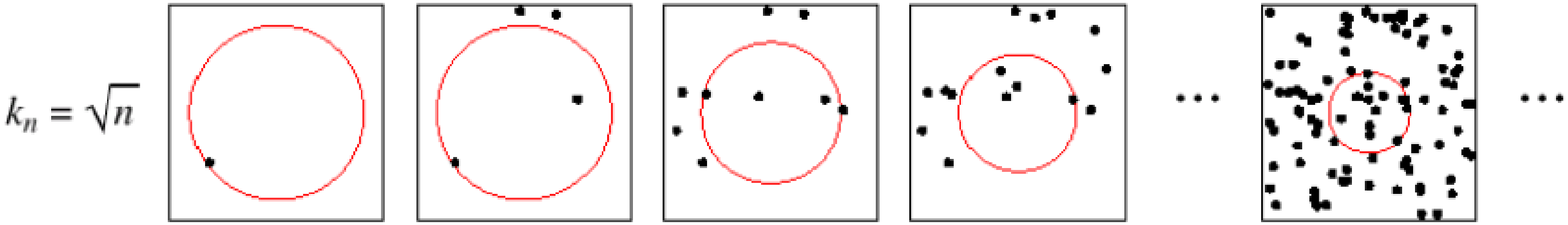
$$\hat{p}(x)=\frac{k}{NV}$$

两种选择策略：

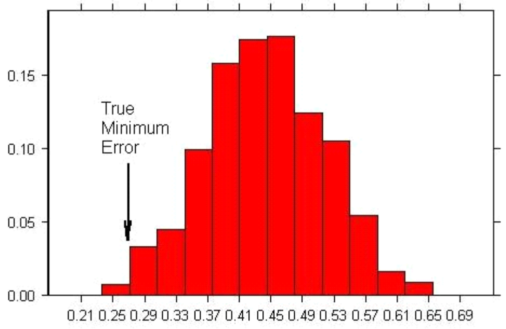
（1）选择 V_n （比如 $V_n=\frac{1}{\sqrt{n}}$ ），同时对 k_n 和 $\frac{k_n}{n}$ 加限制以保证收敛— Parzen窗法



（2）选择 k_n （比如 $k_n=\sqrt{n}$ ），V_n 为正好包含 x 的 k_n 个近邻— k_N 近邻估计



4.4 Parzen窗法



概率密度估计： $\hat{p}(x)=\frac{k}{NV}=\frac{1}{N}\bullet\frac{k}{V}$

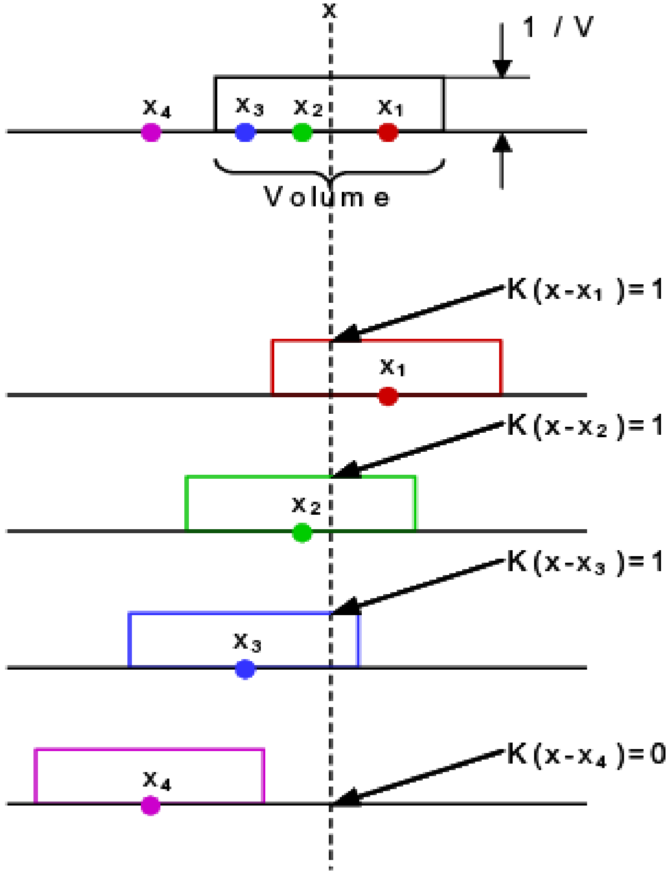
设 x 是 d 维特征向量，每个小舱是一个超立方体，每一维的棱长是 h，则小舱体积为$V=h^d$

核函数（窗函数）： $k(x,x_i)=\frac{1}{V}\varphi(\frac{x-x_i}{h})$

- 其反映了一个观测样本对在x处的概率密度估计的贡献。

则概率密度估计为： $\hat{p}(x)=\frac{1}{N}\sum_{i=1}^{N}k(\mathbf{x},\mathbf{x}_i)$

- 即在每一点上把所有观测样本的贡献平均。



* 注意到核函数估计和直方图法很相似，但窗的位置是由数据来确定的

窗函数条件： $k(x,x_i)\geq 0$ $\int k(x,x_i)dx=1$

4.4 Parzen窗法

常用窗函数：

（1）超立方体窗（方窗）

$$k(x,x_i)=\begin{cases}\dfrac{1}{h^d} & \text{if } \left|x^i-x_i^j\right|\le h/2,\ j=1,2,\cdots,d\\ 0 & \text{otherwise}\end{cases}$$

h 为超立方体棱长，

$V=h^d$

（2）正态窗（高斯窗）

$$k(x,x_i)=\frac{1}{\sqrt{(2\pi)^d\rho^{2d}\,|Q|}}\exp\left\{-\frac{1}{2}\frac{(x-x_i)^TQ^{-1}(x-x_i)}{\rho^2}\right\}$$

$(\Sigma=\rho^2Q)$

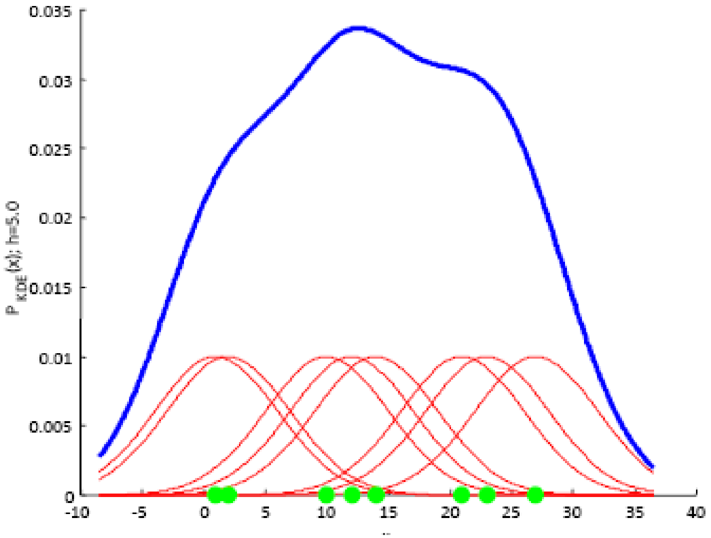
（3）超球窗

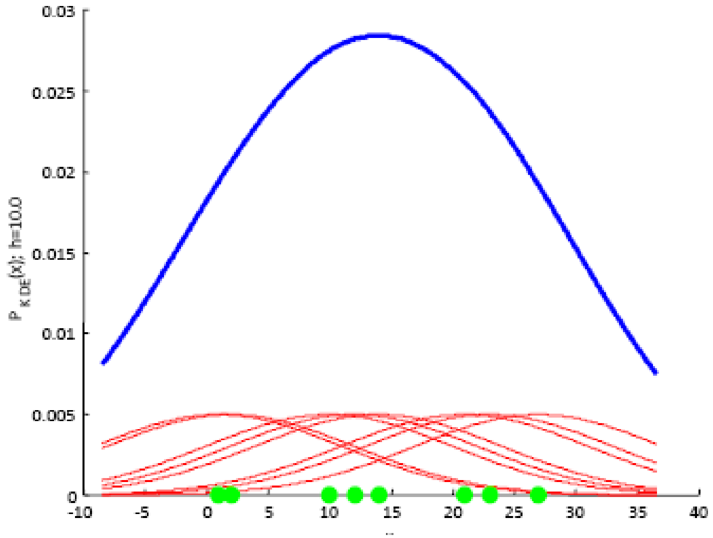
$$k(x,x_i)=\begin{cases}V^{-1} & \text{if } \left\|x-x_i\right\|\le\rho\\ 0 & \text{otherwise}\end{cases}$$

（ V 超球体积，ρ 半径 ）

ρ 的选择：样本数少则选大些，样本数多则选小些。例如： $\rho=N^{-\eta/d}$ $\eta\in(0,1)$

4. 4 Parzen窗法





窗长度ρ对概率密度估计值$p_N(x)$的影响：

- 若ρ太大, $p_N(x)$是$p(x)$的一个平坦、分辨率低的估计, 有平均误差。
- 若ρ太小, $p_N(x)$是$p(x)$的一个不稳定的起伏大的估计,有噪声误差。

4.4 Parzen窗法

Parson窗估计的性质：在满足一定的条件下，估计量 $\hat{p}_N(x)$ 是渐近无偏和平方误差一致的。

其条件是：

- 总体密度 $p(x)$ 在 x 点连续；
- 窗函数满足以下条件：

$k(u) \ge 0, \quad \int k(u)du = 1$ ： 窗函数具有密度函数的性质

$\sup\limits_{u} k(u) < \infty$ ： 窗函数有界

$\lim\limits_{\|u\| \to \infty} k(u) \prod\limits_{i=1}^{d} u_i = 0$ ： 窗函数随着距离的增大很快趋于零

- 窗宽受以下条件约束：

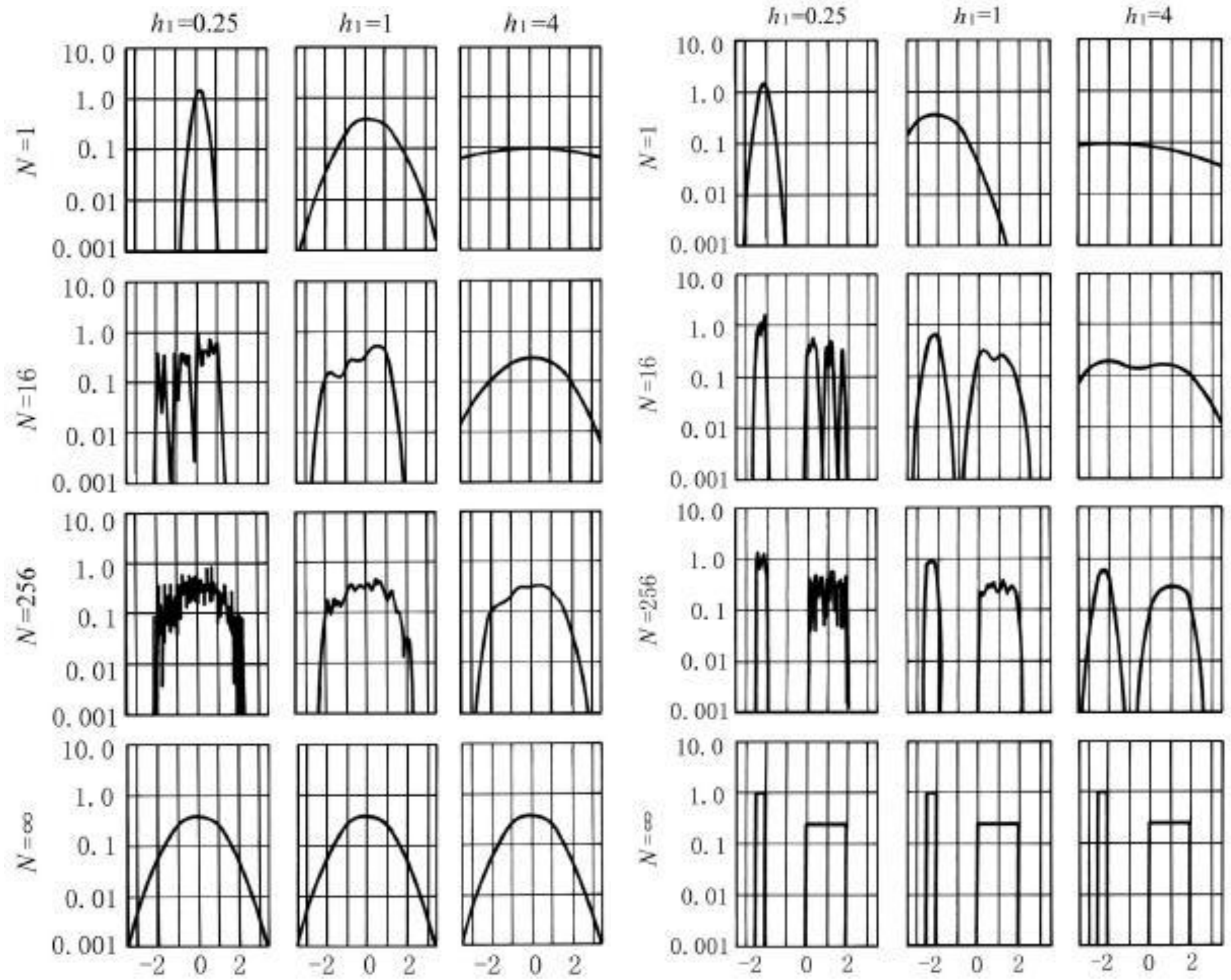
$\lim\limits_{N \to \infty} V_N = 0$ ： 窗体积随着N的增大而趋于零

$\lim\limits_{N \to \infty} NV_N = \infty$ ： 但体积减小的速度要低于1/N

4.4 Parzen窗法

举例：

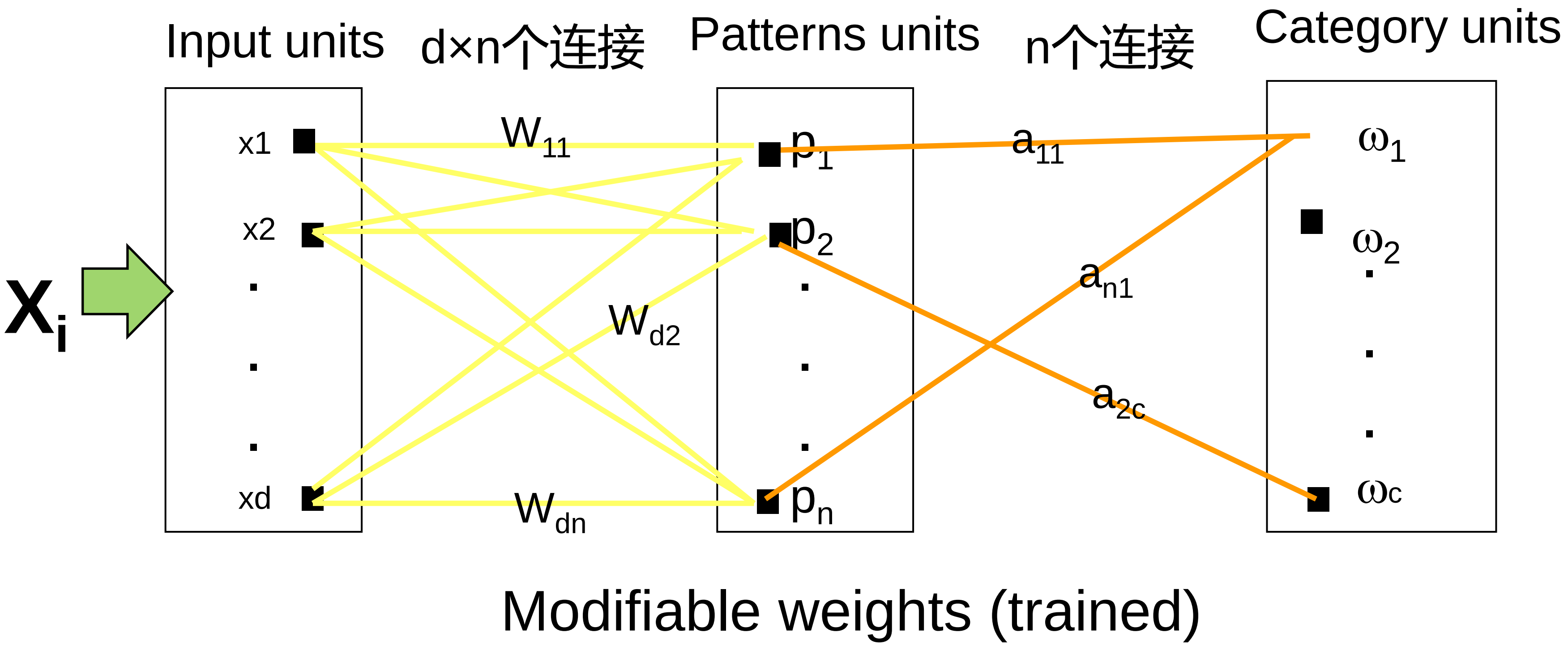
用已知的密度函数产生一系列样本，根据这些样本用Parzen窗法估计概率密度函数，与真实密度函数比较，分析样本数，窗宽等对估计结果的影响。



4.4 Parzen窗法

概率神经网络（*PNN*）：一种Parzen窗的实现

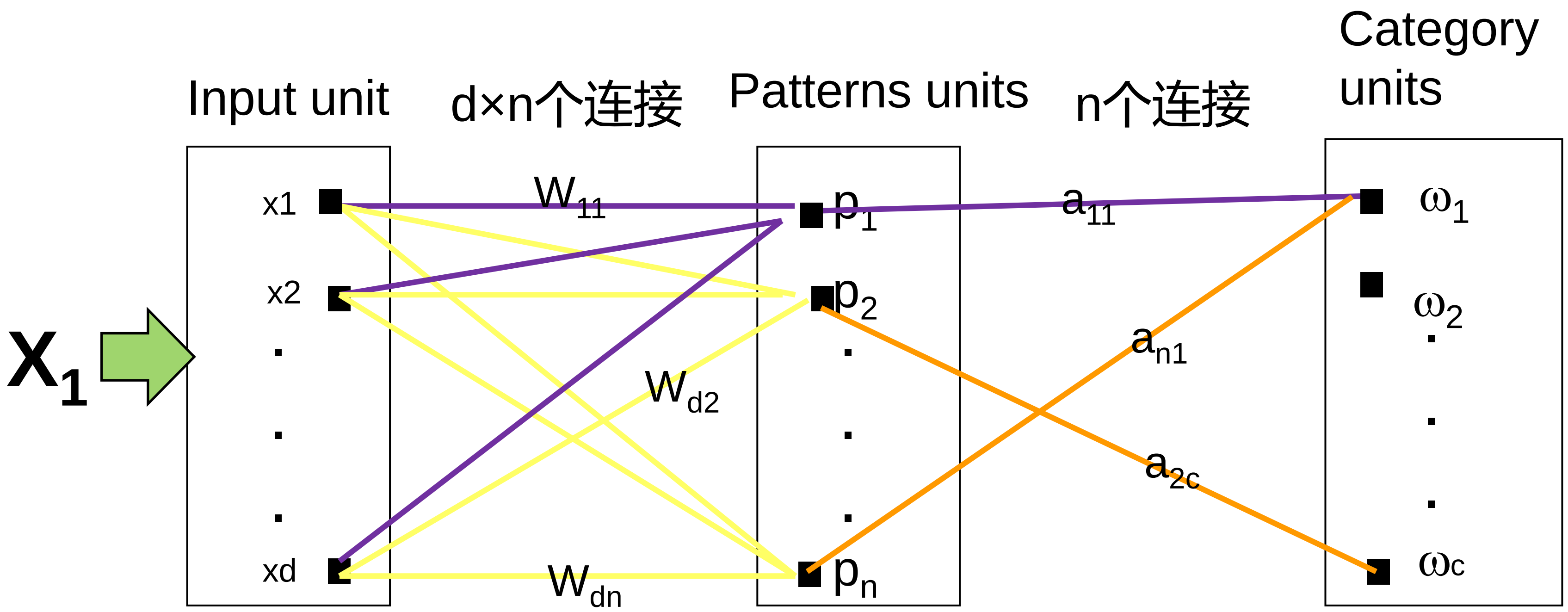
- Compute a Parzen estimate based on n patterns
- Patterns with d features sampled from c classes
- The input unit is connected to n patterns



4. 4 Parzen窗法

概率神经网络（*PNN*）：一种Parzen窗的实现

- Compute a Parzen estimate based on *n* patterns
- Patterns with *d* features sampled from *c* classes
- The input unit is connected to *n* patterns

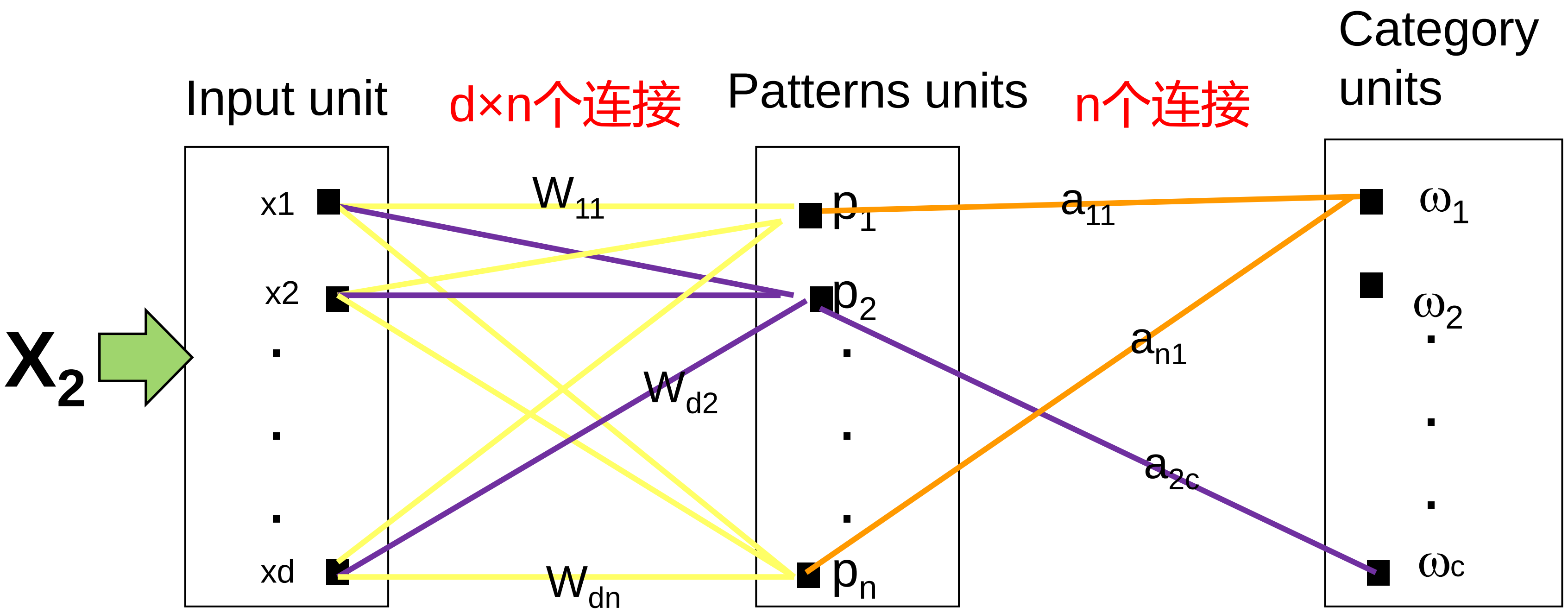


Modifiable weights (trained)

4.4 Parzen窗法

概率神经网络（*PNN*）：一种Parzen窗的实现

- Compute a Parzen estimate based on *n* patterns
- Patterns with *d* features sampled from *c* classes
- The input unit is connected to *n* patterns



Modifiable weights (trained)

4.4 Parzen窗法

Normalization 归一化问题

- Patterns are normalized (or scaled) to have unit length $\sum_{i=1}^{d} x_i^2 = 1$
- This is done by replacing each feature value by $x_j \leftarrow \frac{x_j}{\left(\sum_{i=1}^{d} x_i^2\right)^{1/2}}$
- Effect of normalization $x^t x = 1$

Normalization Example

Example：Normalize x = $[3\ 4]^t$

$(9+16)^{1/2}$ = 5

Normalized vector is =$[3/5\ 4/5]^t$= $[0.6\ 0.8]^t$

Effect of normalization

$$\sum_{i=1}^{d} x_i^2 = 0.36 + 0.64 = 1$$

$$x^t x = \begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix} \begin{bmatrix} 0.6 & 0.8 \end{bmatrix} = 1$$

4.4 Parzen窗法

PNN training

Algorithm 1 (PNN training)

1 begin initialize $k \leftarrow 0$, n = # patterns,

$a_{ki} \leftarrow 0$ for k=1,…,N; i=1,…,c; x = test pattern

2 do $k \leftarrow k+1$

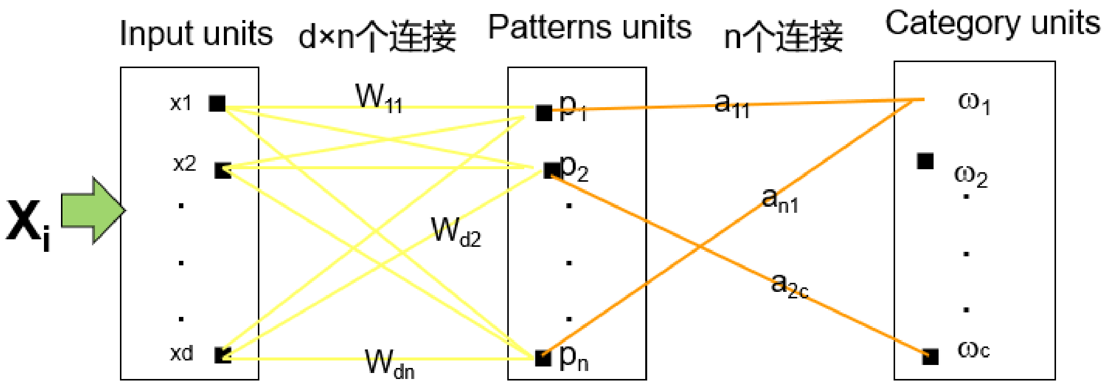
3 $normalize: x_{jk} \leftarrow x_{jk} \Big/ \left(\sum x_{jk}^2\right)^{1/2}, j=1,\ldots,d$

4 $train: \ w_{jk} \leftarrow x_{jk}$

5 $if \quad x \in \omega_i \quad then \quad a_{ki} \leftarrow 1$

6 $until \quad k = N$

7 *end*



a_{ki}对应于被标示的样本和相应的类之间之间的连接

4.4 Parzen窗法

Activation Function 激活函数

$$k(x,x_i)=\frac{1}{\sqrt{(2\pi)^d\rho^{2d}|Q|}}\exp\left\{-\frac{1}{2}\frac{(x-x_i)^TQ^{-1}(x-x_i)}{\rho^2}\right\}$$

Desired Gaussian

$$\varphi\left(\frac{\mathrm{x}-\mathrm{w}_k}{h_N}\right)\propto e^{-(\mathrm{x}-\mathrm{w}_k)^t(\mathrm{x}-\mathrm{w}_k)/2\sigma^2}$$

样本对x的贡献：权重对应样本的特征值

$$e^{-(\mathrm{x}-\mathrm{w}_k)^t(\mathrm{x}-\mathrm{w}_k)/2\sigma^2}$$

$$=e^{-\left(\mathrm{x}^t\mathrm{x}+\mathrm{w}_k^t\mathrm{w}_k-2\mathrm{x}^t\mathrm{w}_k\right)/2\sigma^2}$$

$$=e^{(net_k-1)/\sigma^2}$$

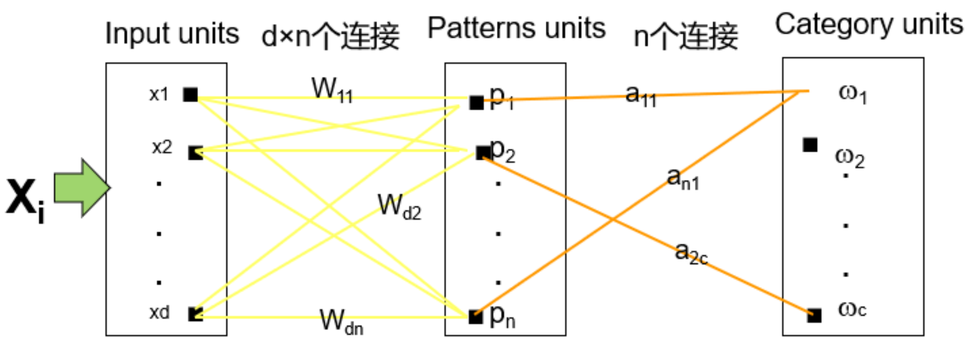
Simplified form due to normalization1

$$\mathrm{x}^t\mathrm{x}=\mathrm{w}_k^t\mathrm{w}_k=1$$

$$net_k=\mathrm{w}_k{}^t\mathrm{x}$$

4. 4 Parzen窗法

PNN classification



1. 归一化待分类实例*x;*
2. 对每个模式计算内积 $net_k = w_k^t . x$
3. 在有连接的输出层上累加

$$g_i = g_i + \exp\left[\frac{net_k - 1}{\sigma^2}\right]$$

4.最大的响应类别做为最后分类结果

Algorithm 2 (PNN classification)

1 begin initialize k = 0; x = test pattern

2 do $k \leftarrow k+1$

3 $\mathrm{net}_k \leftarrow \mathrm{w}_k^t x$

4 if $\mathrm{a}_{ki} = 1$ *then* $g_i \leftarrow g_i + \exp\left[(net_k - 1)/\sigma^2\right]$

5 until $k = N$

6 *return* $class \leftarrow \arg\max_i \; g_i(x)$

7 *end*

选用合适的激活函数

每个类上的输出是对应于标记为该类别的样本的激活函数g_i的总和,选最大的**和值**对应的类作为判决结果.

Ending