第4章：非参数方法

刘成林(liucl@nlpr.ia.ac.cn)

2016年10月12日

助教：杨学行(xhyang@nlpr.ia.ac.cn)
吴一超(yichao.wu@nlpr.ia.ac.cn)

中国科学院大学
University of Chinese Academy of Sciences

上次课主要内容回顾

- 特征维数与过拟合
  - 增加特征带来更多判别信息
  - 克服过拟合的方法？
- 期望最大法(EM)
  - 对数似然度对缺失数据的期望
  - EM for Gaussian mixture
- 隐马尔可夫模型(HMM)
  - Three basic problems
  - Viterbi Algorithm (DP)
  - Extensions

提纲

- 第4章 非参数方法
  - 密度估计
  - Parzen窗方法
  - K近邻估计
  - 最近邻规则
  - 距离度量
  - Reduced Coulomb Energy Network
  - Approximation by Series Expansion

密度估计

- 概率和密度
  - 概率：特征空间中一定区域内样本的比率

$$P = \int_{\mathcal{R}} p(\mathbf{x}')\, d\mathbf{x}'$$

  - 假设局部区域（体积为V，样本数k）内等概率密度

$$\int_{\mathcal{R}} p(\mathbf{x}')\, d\mathbf{x}' \simeq p(\mathbf{x})V \qquad p(\mathbf{x}) \simeq \frac{k/n}{V}$$

  - 如何决定局部区域的大小：随样本数*n*变化
  - $p_n(\mathbf{x})$收敛到$p(\mathbf{x})$的条件

$$\lim_{n\to\infty} V_n = 0$$

$$\lim_{n\to\infty} k_n = \infty$$

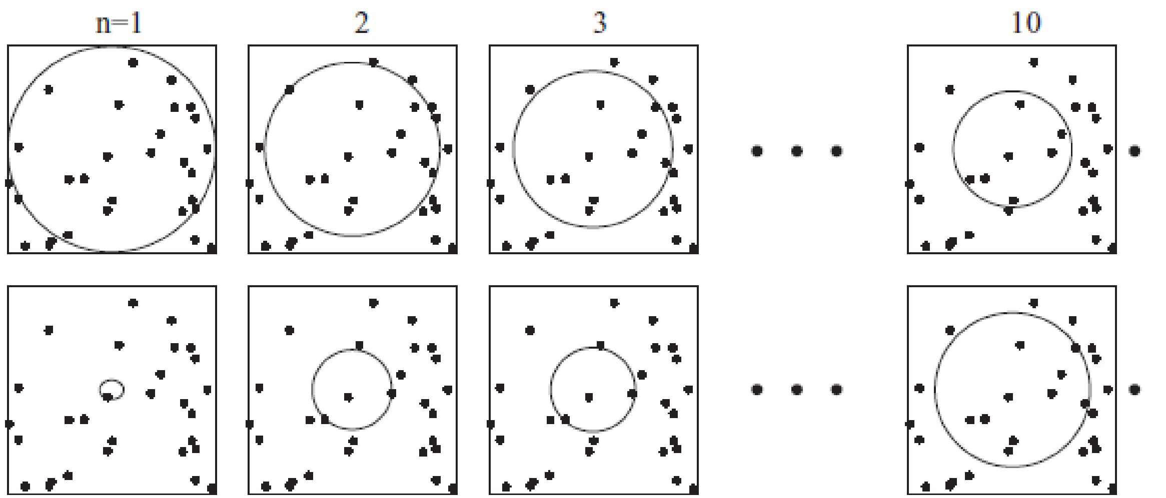
$$\lim_{n\to\infty} k_n/n = 0$$

University of Chinese Academy of Sciences

- 非参数概率密度估计
  - Parzen window: 固定局部区域体积*V*，*k*变化
  - k-nearest neighbor: 固定局部样本数*k*，*V*变化

Parzen window
$V_n = 1/\sqrt{n}$

k-NN
$k_n = \sqrt{n}$

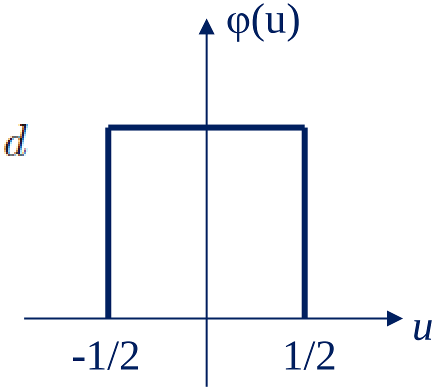


（这里*n*不一定指样本数）

Parzen Window

- 窗函数：hypercube

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq 1/2 \qquad j = 1, \ldots, d \\ 0 & \text{otherwise.} \end{cases}$$



  - 满足条件

$$\varphi(\mathbf{x}) \geq 0 \qquad \int \varphi(\mathbf{u})\, d\mathbf{u} = 1$$

  - 以x为中心、体积为 $V_n = h_n^d$ 的局部区域内样本数

$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

  - 概率密度估计k_n/nV_n

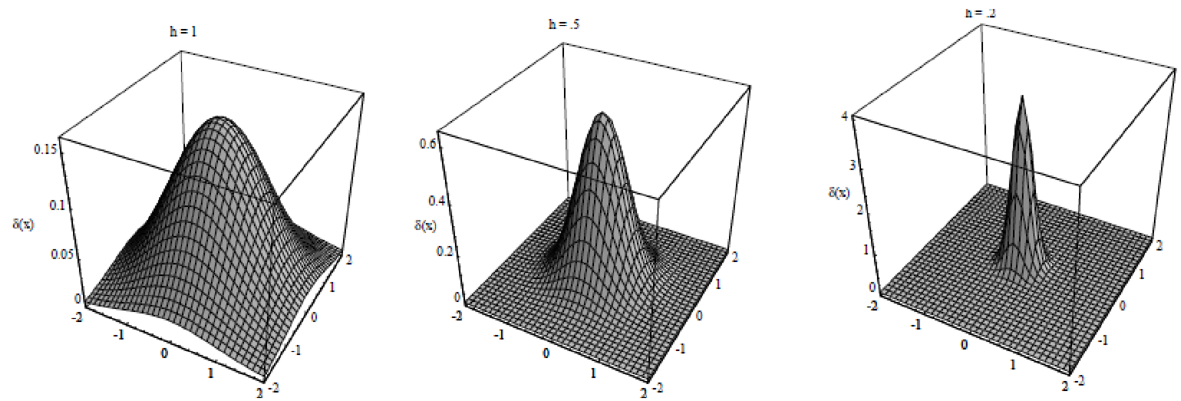
$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

- 推广：满足密度函数要求的窗函数，如高斯函数

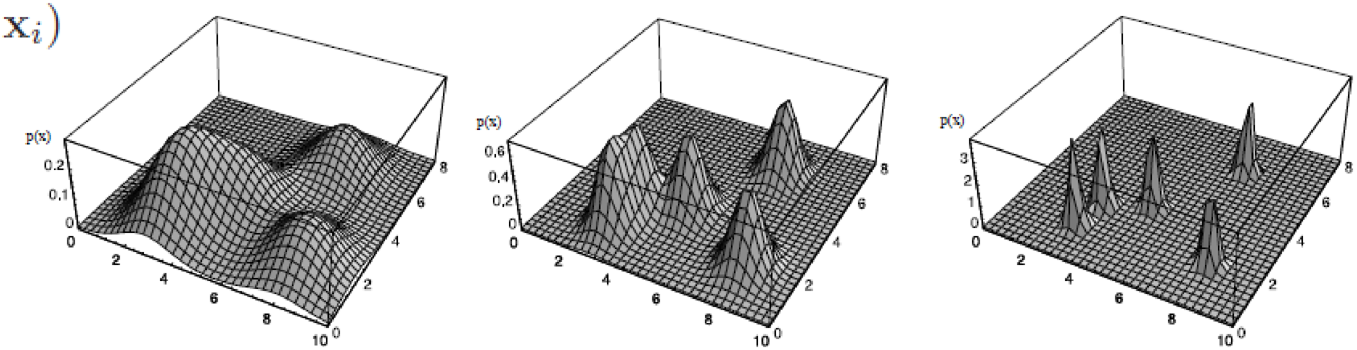
$$\varphi(\mathbf{x}) \geq 0 \qquad \int \varphi(\mathbf{u})\, d\mathbf{u} = 1$$

Gaussian window, variable width (*h*=1, 0.5, 0.2)

$$\delta_n(\mathbf{x}) = \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$



$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\delta_n(\mathbf{x}-\mathbf{x}_i)$$



Large *h*: low variability, under fitting
Small *h*: high variability, overfitting

- Parzen窗密度估计的收敛性
  - $p_n(\mathbf{x})$的期望是$p(\mathbf{x})$的平滑（卷积）
    - Samples $\mathbf{x}_1,\ldots,\mathbf{x}_n$ are i.i.d from $p(\mathbf{x})$

$$\begin{aligned}
\bar{p}_n(\mathbf{x}) &= \mathcal{E}[p_n(\mathbf{x})] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathcal{E}\Big[\frac{1}{V_n}\varphi\Big(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\Big)\Big] \\
&= \int \frac{1}{V_n}\varphi\Big(\frac{\mathbf{x}-\mathbf{v}}{h_n}\Big)\,p(\mathbf{v})\,d\mathbf{v} \\
&= \int \delta_n(\mathbf{x}-\mathbf{v})p(\mathbf{v})\,d\mathbf{v}.
\end{aligned}$$

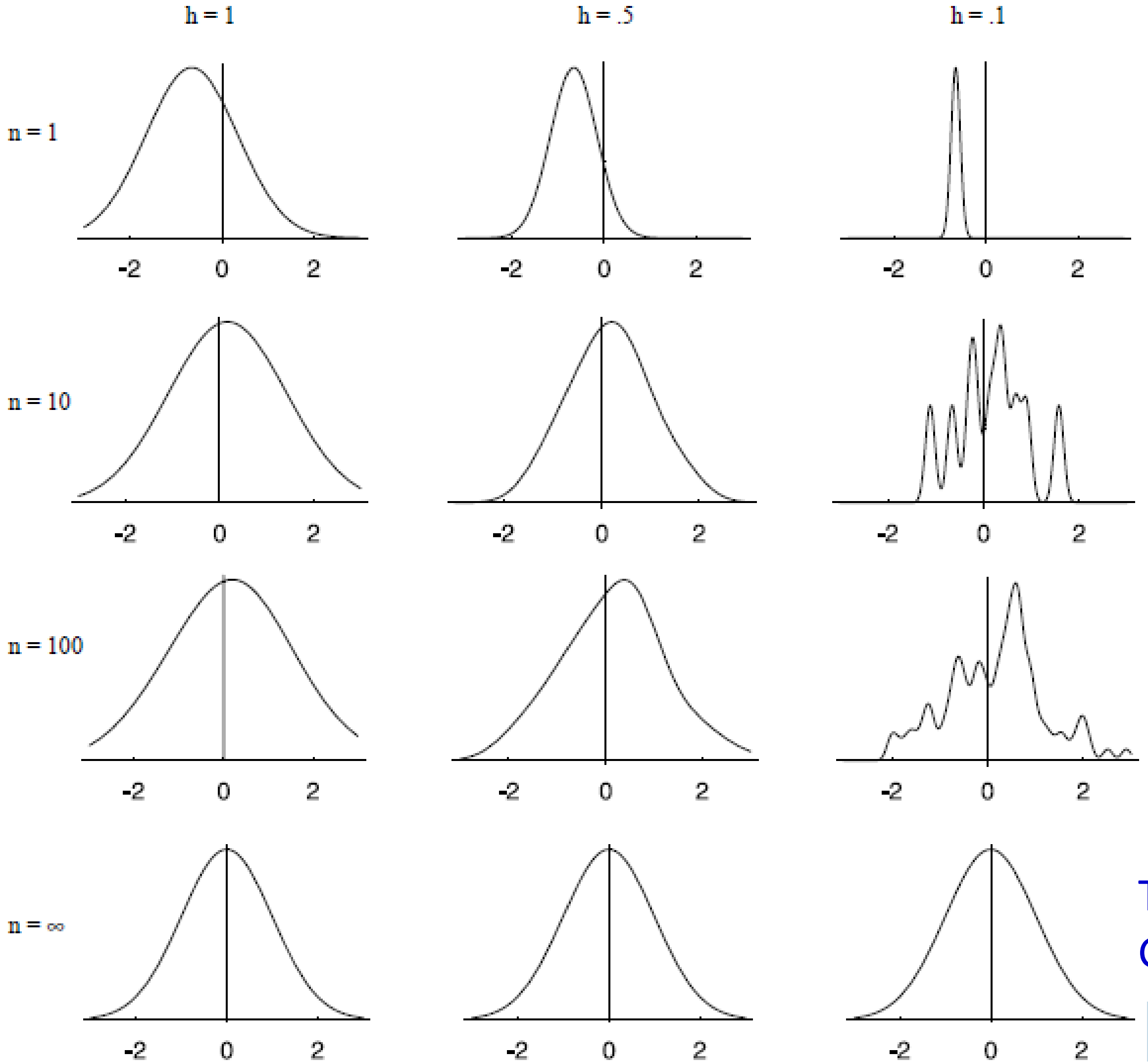
When $n \to \infty$ $\lim_{n\to\infty} V_n = 0$ $\lim_{n\to\infty} nV_n = \infty$

$$\bar{p}_n(\mathbf{x}) \to p(\mathbf{x})$$

University of Chinese Academy of Sciences

- 示例：高斯窗函数 $\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_n}\varphi\left(\frac{x-x_i}{h_n}\right) \qquad h_n = h_1/\sqrt{n}$$



True p(x):
Gaussian

University of Chinese A

2D case

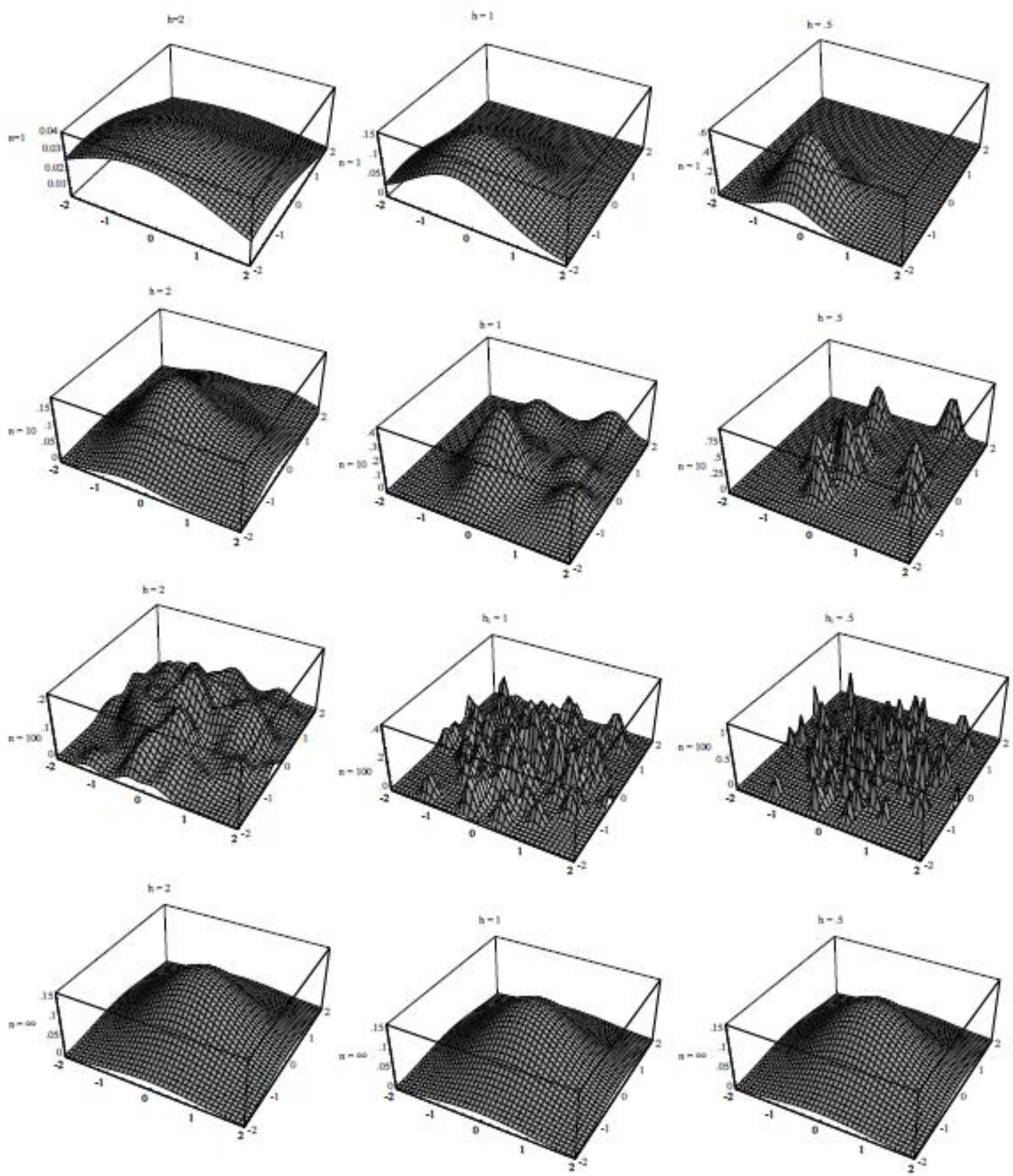
h=2 | *h*=1.0 | *h*=0.5

n=1

n=10

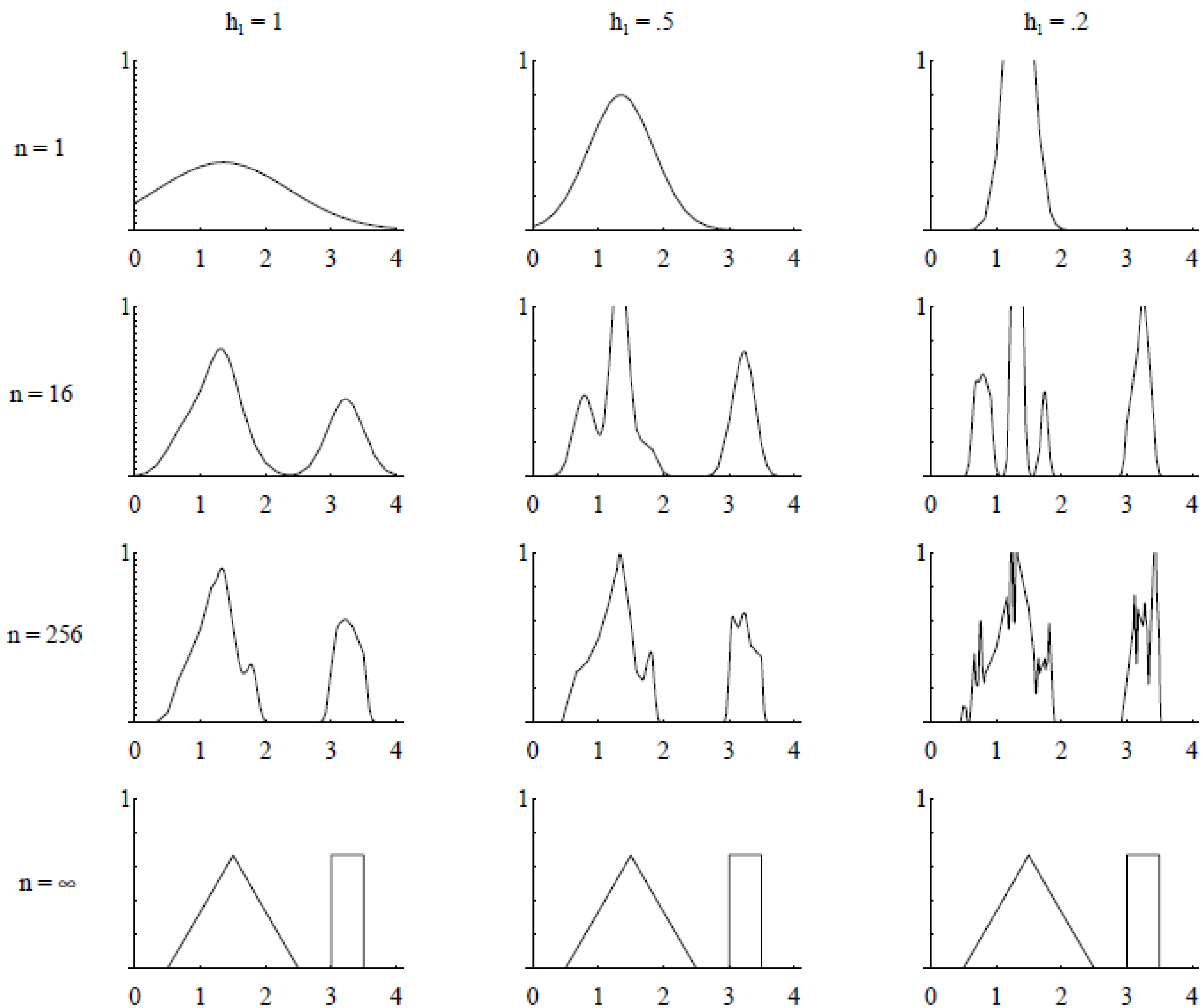
n=100

n=∞



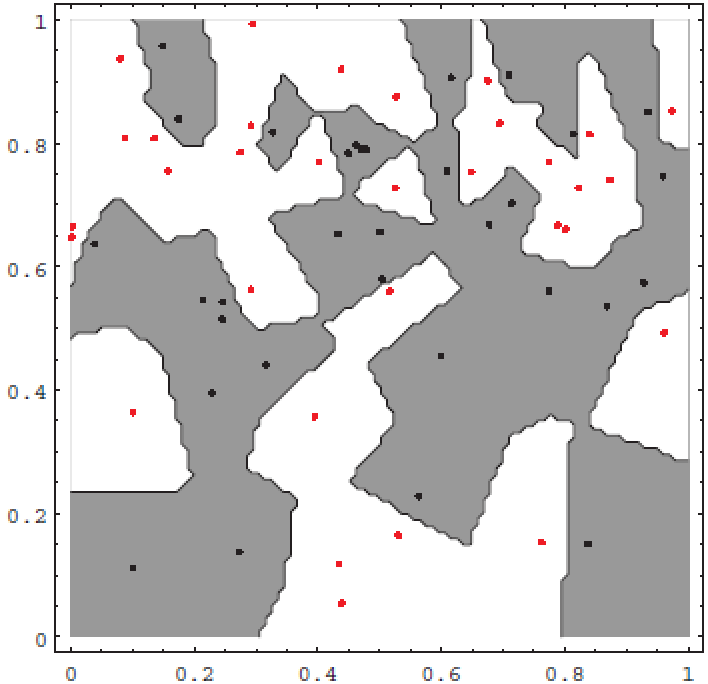
True $p(\mathbf{x})$: Gaussian

- Bimodal distribution



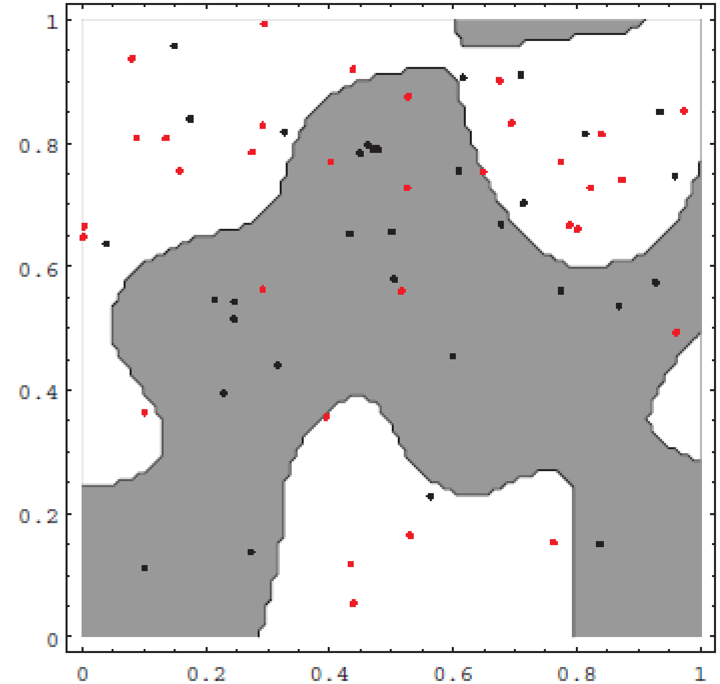
- 分类的例子 $\max_i p(\mathbf{x} \mid \omega_i) P(\omega_i)$

Small *h*



Decision regions

Large *h*

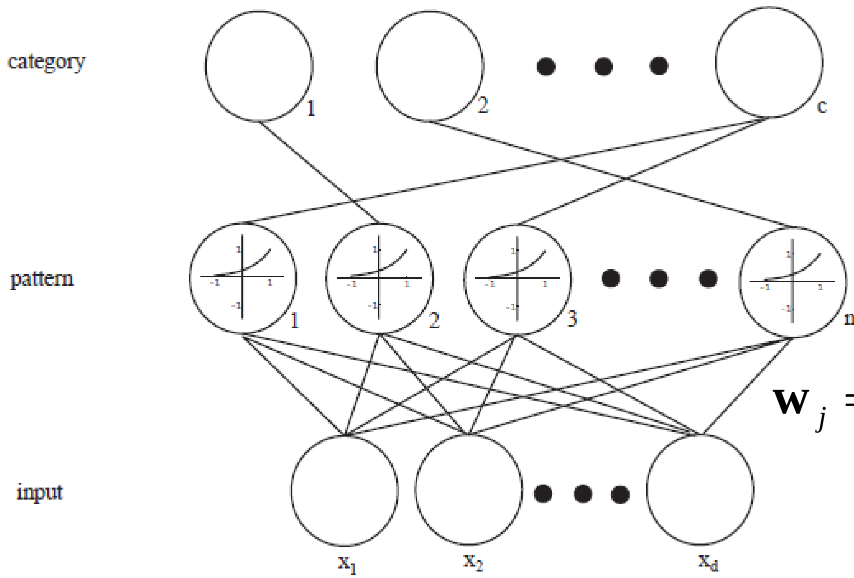


上部和下部密度区别大，适合不同的*h*值
（考虑Generalization）

University of Chinese Academy of Sciences

- 窗宽h_n选择经验
  - 一般原则：n越大或密度越大，h_n越小
  - 随n变化：$V_n = V_1/\sqrt{n}$
  - 随x变化：$h(\mathbf{x}), h(\mathbf{x}_i)$
    - $\mathbf{x}$测试样本，$\mathbf{x}_i$训练样本
    - 比如：根据k-NN的距离判断
  - 交叉验证(cross validation)
    - 比如选择V_1

- Probabilistic Neural Network (PNN)
  - 输出每个类别的概率密度
  - 隐节点：pattern unit, 对应窗函数
  - Normalized pattern: $\mathbf{x} \leftarrow \mathbf{x}/\|\mathbf{x}\|$



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

$$z_k = \mathbf{w}_k^t \mathbf{x}$$

$$\mathbf{w}_j = \mathbf{x}_j$$

Why $e^{(z_k-1)/\sigma^2}$

$$\varphi\left(\frac{\mathbf{x}_k - \mathbf{w}_k}{h_n}\right) \propto \overbrace{e^{-(\mathbf{x}-\mathbf{w}_k)^t(\mathbf{x}-\mathbf{w}_k)/2\sigma^2}}^{\text{desired Gaussian}}$$

$$= e^{-(\mathbf{x}^t\mathbf{x}+\mathbf{w}_k^t\mathbf{w}_k-2\mathbf{x}^t\mathbf{w}_k)/2\sigma^2} = e^{(z_k-1)/\sigma^2}$$

K近邻估计

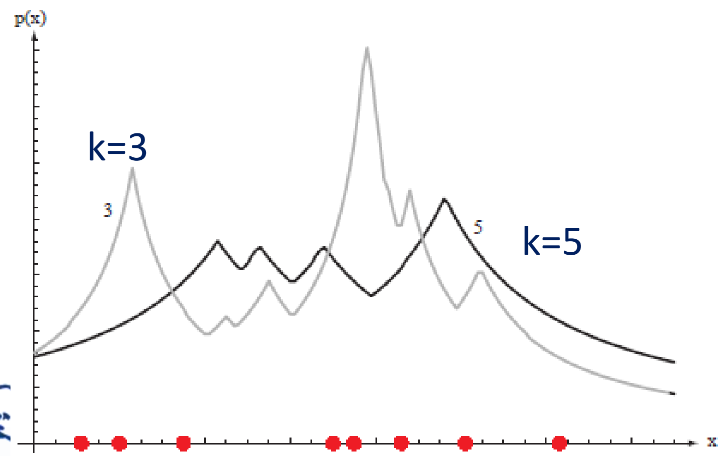
- 概率密度估计
  - 固定局部区域样本数*k*, 体积*V*变化

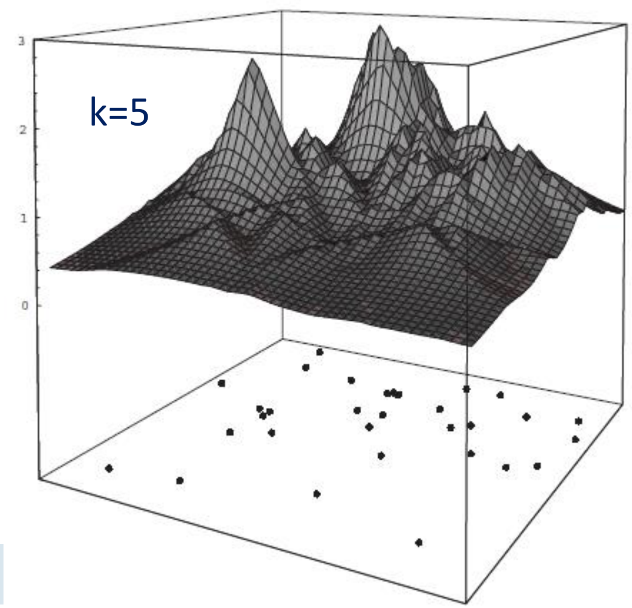
$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$

  - 收敛到*p*(**x**)条件

$$\lim_{n\to\infty} k_n = \infty \text{ and } \lim_{n\to\infty} k_n/n = 0$$

  - 一种选择：$k_n = \sqrt{n} \quad V_n \simeq 1/(\sqrt{n}p(\mathbf{x}))$
  - 1D, 2D的例子

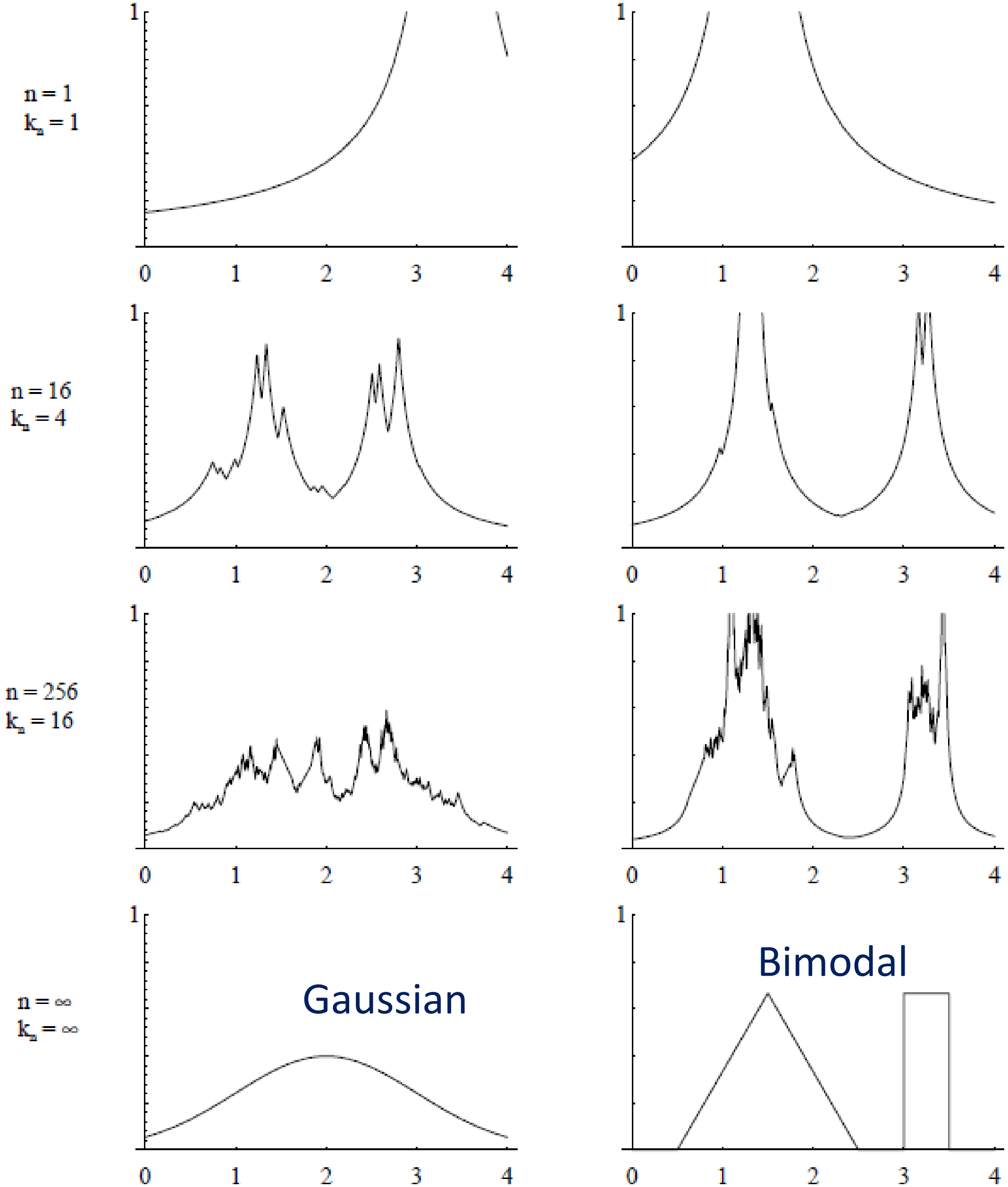




More 1D examples

$$k_n = \sqrt{n}$$

$$p_n(x) = \frac{\sqrt{n}/n}{2|x - x_{kNN}|}$$



True $p(\mathbf{x})$

- K-NN分类：后验概率
  - k_i NNs from class i $\quad k = \sum_{i=1}^{c} k_i$

$$p_n(\mathbf{x}, \omega_i) = \frac{k_i/n}{V}$$

$$\boxed{P_n(\omega_i|\mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_{j=1}^{c} p_n(\mathbf{x}, \omega_j)} = \frac{k_i}{k}}$$

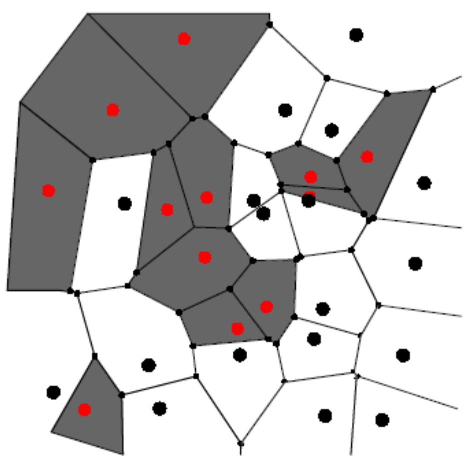
  - 分类错误率：当 $\lim_{n\to\infty} k_n = \infty$ and $\lim_{n\to\infty} k_n/n = 0$ 趋近贝叶斯错误率

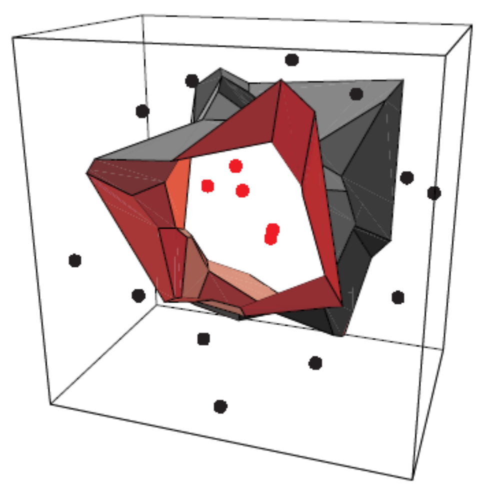
最近邻规则

- Nearest Neighbor (1-NN) Rule
  - Among labeled data $\mathcal{D}^n = \{\mathrm{x}_1, ..., \mathrm{x}_n\}$, **x'** is the NN of **x**
  - Assume $P(\omega|\mathrm{x}') \simeq P(\omega_i|\mathrm{x})$
  - Classification: MAP

$$\omega_m = \arg\max_i P(\omega_i \mid \mathbf{x}) = \omega(\mathbf{x}')$$

  - Decision regions: Voronoi tesselation





- 最近邻规则的错误率

$$P(e) = \int P(e|\mathbf{x})p(\mathbf{x})\ d\mathbf{x}$$

$$P(e|\mathbf{x}) = \int \underline{P(e|\mathbf{x},\mathbf{x}')}p(\mathbf{x}'|\mathbf{x})\ d\mathbf{x}'$$

x': NN of **x**

  - When n→∞, *p*(**x**'|**x**) approaches delta function centered at **x**
  - For *P*(*e*|**x**,**x**'), assume **x** and $\mathbf{x}_n$' (nearest training sample, independent) are associated with class variables θ and θ_n', respectively

$$P(\theta, \theta'_j|\mathbf{x}, \mathbf{x}'_j) = P(\theta|\mathbf{x})P(\theta'_j|\mathbf{x}'_j)$$

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

$$\lim_{n\to\infty} P_n(e|\mathbf{x}) = \int \left[1 - \sum_{i=1}^{c} P(\omega_i|\mathbf{x})P(\omega_i|\mathbf{x}')\right]\underline{\delta(\mathbf{x}'-\mathbf{x})}\ d\mathbf{x}' = 1 - \sum_{i=1}^{c} P^2(\omega_i|\mathbf{x})$$

  - Asymptotic error rate

$$\begin{aligned} P &= \lim_{n\to\infty} P_n(e) \\ &= \lim_{n\to\infty} \int P_n(e|\mathbf{x})p(\mathbf{x})\ d\mathbf{x} \\ &= \int \left[1 - \sum_{i=1}^{c} P^2(\omega_i|\mathbf{x})\right]p(\mathbf{x})\ d\mathbf{x} \end{aligned}$$

- Error bound of 1-NN rule

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

Minimized when P_i ($i \neq m$) are equal

$$P(\omega_i|\mathbf{x}) = \begin{cases} \frac{P^*(e|\mathbf{x})}{c-1} & i \neq m \\ 1 - P^*(e|\mathbf{x}) & i = m \end{cases}$$

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

(Bayes error)

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

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

- Error rate $P = \int \left[1 - \sum_{i=1}^{c} P^2(\omega_i|\mathbf{x})\right] p(\mathbf{x})\, d\mathbf{x} \longrightarrow P \leq 2P^*$

$$\begin{aligned} \mathrm{Var}[P^*(e|\mathbf{x})] &= \int [P^*(e|\mathbf{x}) - P^*]^2 p(\mathbf{x})\, d\mathbf{x} \\ &= \int P^{*2}(e|\mathbf{x}) p(\mathbf{x})\, d\mathbf{x} - P^{*2} \geq 0 \longrightarrow \int P^{*2}(e|\mathbf{x}) p(\mathbf{x})\, d\mathbf{x} \geq P^{*2} \end{aligned}$$

- Error bound

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

Break

K近邻的快速计算

- 分类的计算复杂度$O(dn)$
- 近邻搜索的三种策略
  - Partial distance
  - Prestructuring
  - Editing (pruning, condensing)

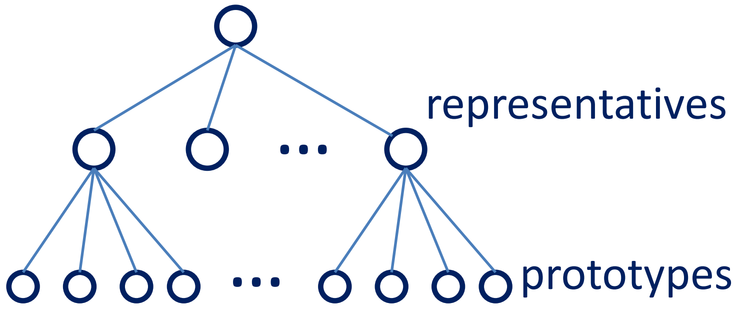
Partial square distance (r<d)

$$D_r^2(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{r}(a_i - b_i)^2$$

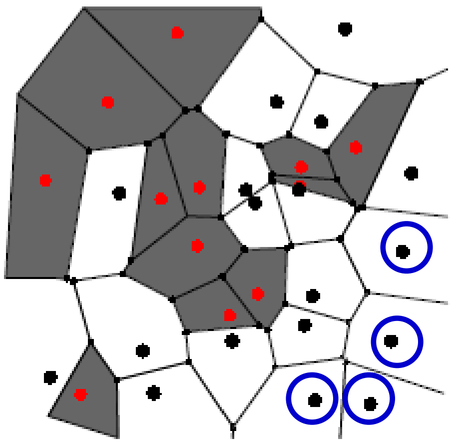
Full distance to the current closest prototype $D^2(\mathbf{x},\mathbf{x}')$

Terminate computing if the partial square distance is greater than $D^2(\mathbf{x},\mathbf{x}')$

- Prestructuring
  - Search tree, prototypes are linked to the nodes, each labeled with a representative prototype
    - Constructed by clustering, e.g.
  - 先找出到x的最近代表点，然后计算与最近代表点连接的原型的距离，找出最近原型
  - 可结合partial distance
  - 为保证找到最近原型，应从多个代表点的原型中搜索



- Editing
  - Remove prototypes that are surrounded by samples (Voronoi neighbors) of same class



距离度量

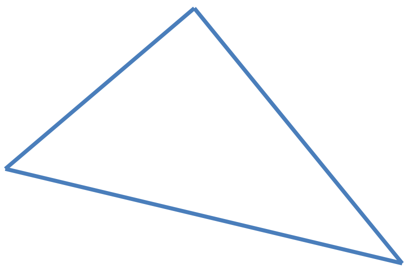
- 距离度量(metric)的性质

  non-negativity: $D(\mathrm{a}, \mathrm{b}) \geq 0$

  reflexivity: $D(\mathrm{a}, \mathrm{b}) = 0$ if and only if $\mathrm{a} = \mathrm{b}$

  symmetry: $D(\mathrm{a}, \mathrm{b}) = D(\mathrm{b}, \mathrm{a})$

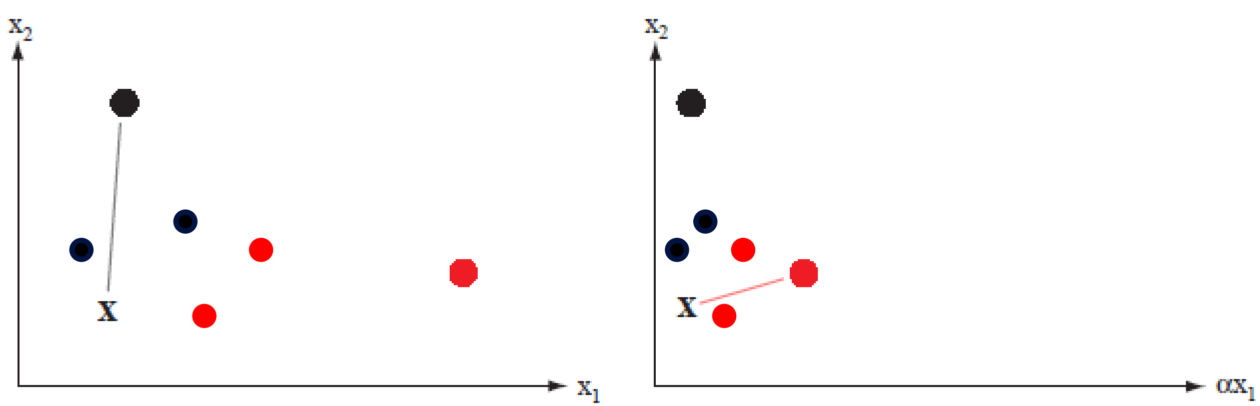
  triangle inequality: $D(\mathrm{a}, \mathrm{b}) + D(\mathrm{b}, \mathrm{c}) \geq D(\mathrm{a}, \mathrm{c})$



Euclidean metric

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

- Metric影响分类性能
  - 比如，当特征变尺度



- 几种Metric
  - Minkowski (L_k norm)

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

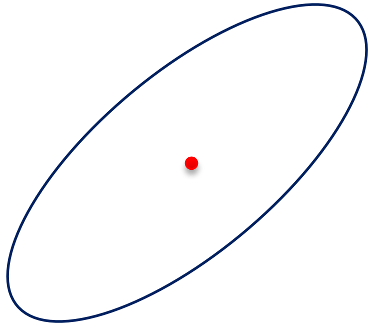
  - Manhattan (city block distance): k=1
  - Tanimoto metric (for binary features)

$$D_{Tanimoto}(\mathcal{S}_1, \mathcal{S}_2) = \frac{n_1 + n_2 - 2n_{12}}{n_1 + n_2 - n_{12}}$$

- Metric Learning
  - Parameters in metric optimized in learning (e.g., empirical risk minimization)

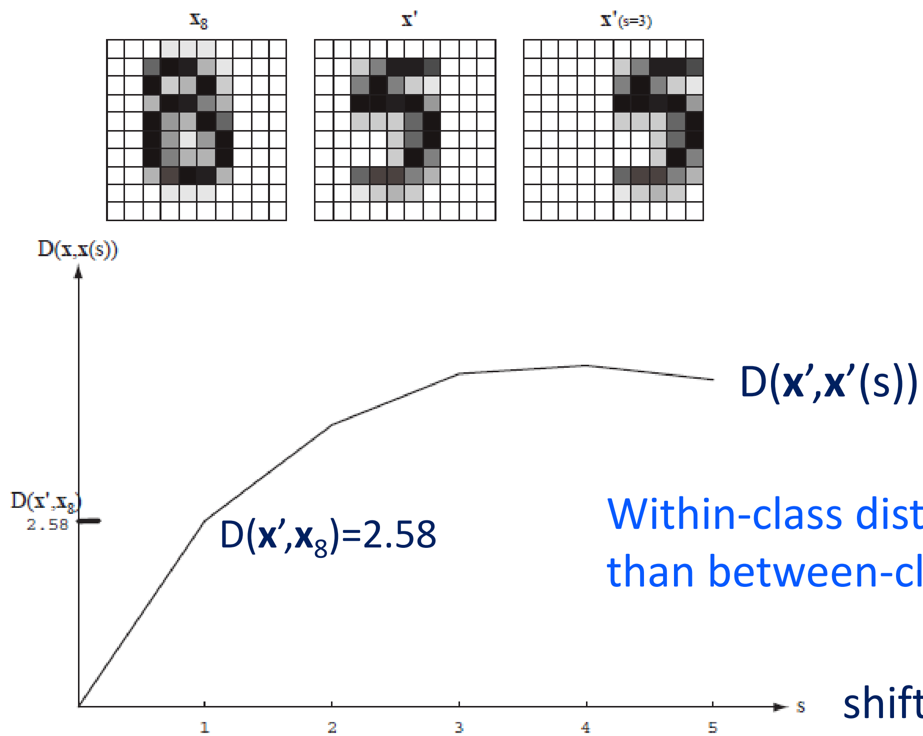
$$D_{\mathbf{w}}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{d} w_i (a_i - b_i)^2$$

$$D_{\Sigma}(\mathbf{a}, \mathbf{b}) = (\mathbf{a} - \mathbf{b})^t \Sigma^{-1} (\mathbf{a} - \mathbf{b})$$



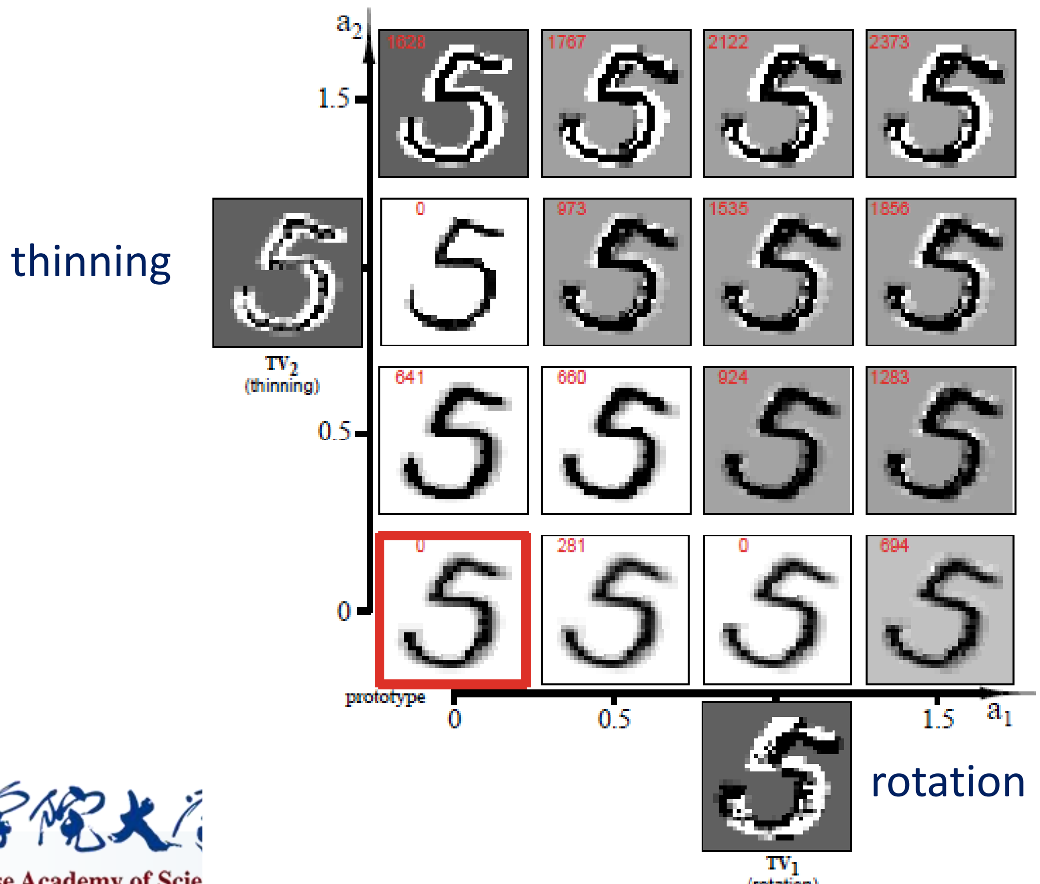
Tangent Distance

- Image Shape Transformation
  - Shift (translation), rotation, scaling, distortion
  - Distance sensitive to transformation



Within-class distance maybe larger than between-class distance

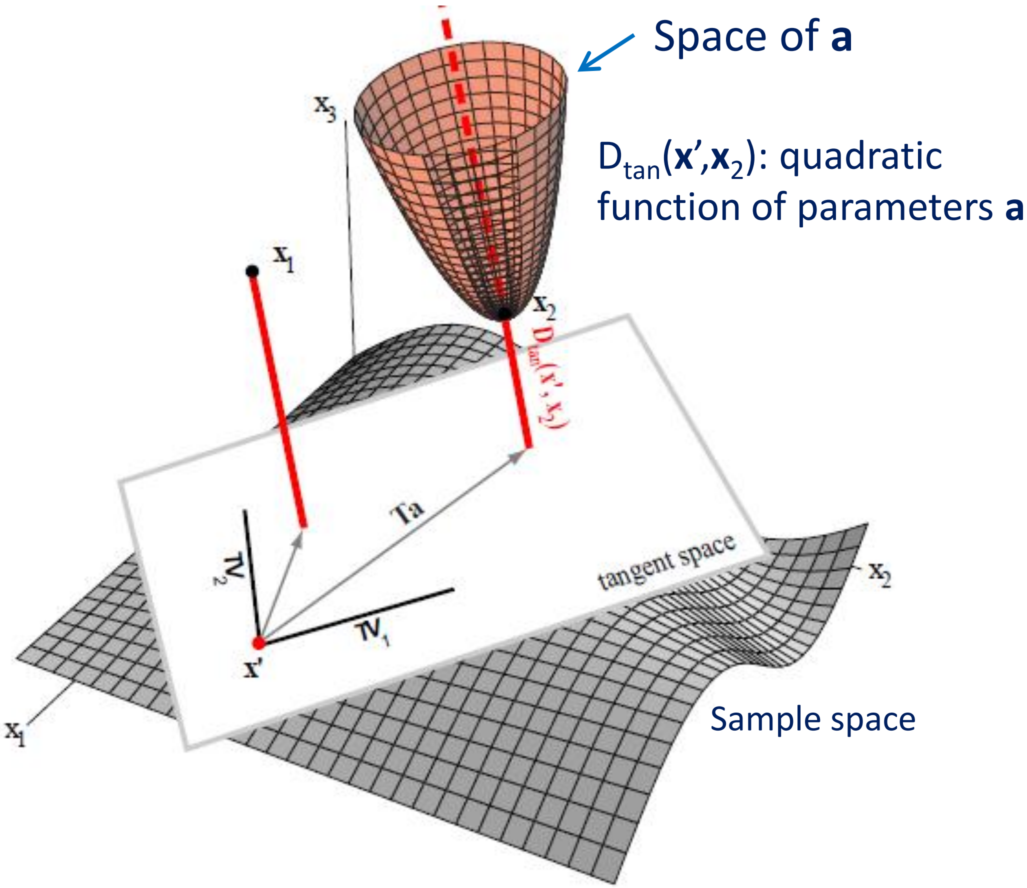
- Tangent distance
  - Search for optimal parameters for a combination of transformations for a prototype to minimize the distance to test sample
  - Parameterized transformation: $\mathcal{F}_i(\mathrm{x}';\ \alpha_i)$
  - Tangent vectors: $\mathbf{TV}_i = \mathcal{F}_i(\mathrm{x}';\ \alpha_i) - \mathrm{x}'$
  - Linear combination in the space spanned by TVs: $\mathrm{x}' + \mathbf{Ta}$



- Tangent distance
  - Euclidean distance to tangent space

$$D_{tan}(\mathrm{x}', \mathrm{x}) = \min_{\mathbf{a}}[\|(\mathrm{x}' + \mathbf{Ta}) - \mathrm{x}\|]$$

  - Optimization: gradient search

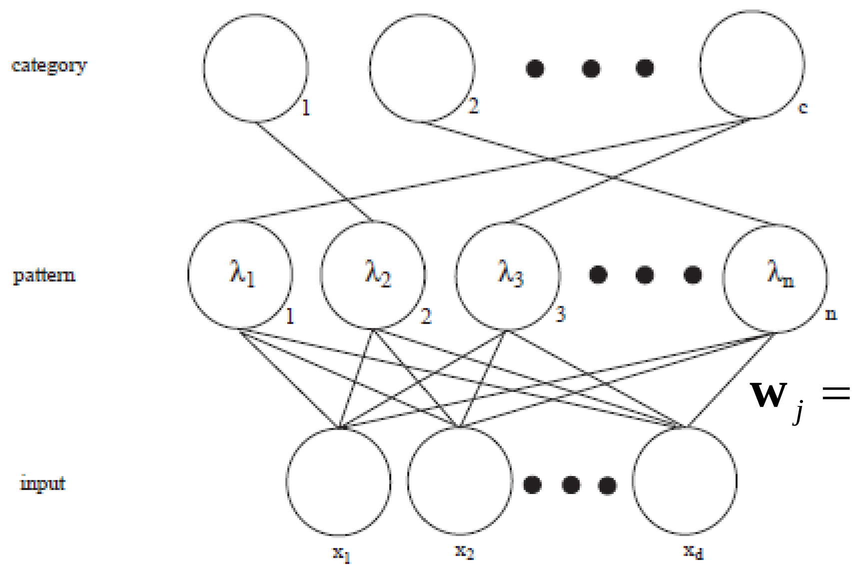


Reduced Coulomb Energy Network

- RCE Network
  - Hidden node (corresponding to a training sample): hypersphere with **radius** according to the distance to nearest point of different class

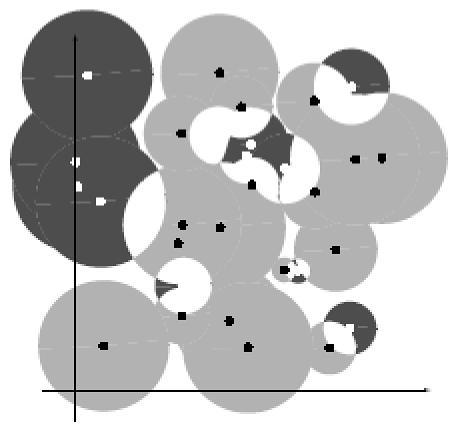
$$\epsilon = \text{small param}, \lambda_m = \text{max radius}$$

$$\lambda_j \leftarrow \min[\min_{\mathbf{x} \notin \omega_i} D(\mathbf{x}, \mathbf{x}') - \varepsilon, \lambda_m]$$



$\mathbf{w}_j = \mathbf{x}_j$ (normalized)

- RCE分类规则
  - 找出包含x的隐节点（超球体），如果这些节点的类别标号一致，则分类到这个类别
    - 没有节点包含x，或者类别不一致（不同类别超球体重叠）的情况，则拒识



白色区域：
ambiguous

Approximation by Series Expansion

- Parzen窗密度估计：计算量大
- 窗函数用序列展开

$$\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) = \sum_{j=1}^{m} a_j \psi_j(\mathbf{x}) \chi_j(\mathbf{x}_i)$$

$$\sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) = \sum_{j=1}^{m} a_j \psi_j(\mathbf{x}) \sum_{i=1}^{n} \chi_j(\mathbf{x}_i)$$

$$p_n(\mathbf{x}) = \sum_{j=1}^{m} b_j \psi_j(\mathbf{x}) \qquad b_j = \frac{a_j}{nV_n} \sum_{i=1}^{n} \chi_j(\mathbf{x}_i)$$

  - b_j可离线计算，p_n(x)只需m次计算($m<n$)

- 高斯窗函数的Taylor展开

$$\sqrt{\pi}\,\varphi(u) = e^{-u^2} \simeq \sum_{j=0}^{m-1} (-1)^j \frac{u^{2j}}{j!}$$

m=2

$$\begin{aligned}\sqrt{\pi}\varphi\left(\frac{x-x_i}{h}\right) &\simeq 1-\left(\frac{x-x_i}{h}\right)^2 \\ &= 1+\frac{2}{h^2}x\,x_i-\frac{1}{h^2}\,x^2-\frac{1}{h^2}\,x_i^2\end{aligned}$$

$$\boxed{\sqrt{\pi}p_n(x) = \frac{1}{nh}\sum_{i=1}^{n}\sqrt{\pi}\varphi\left(\frac{x-x_i}{h}\right) \simeq b_0+b_1x+b_2x^2}$$

$$b_0 = \frac{1}{h}-\frac{1}{h^3}\frac{1}{n}\sum_{i=1}^{n}x_i^2 \qquad b_1 = \frac{2}{h^3}\frac{1}{n}\sum_{i=1}^{n}x_i \qquad b_2 = -\frac{1}{h^3}$$

只有当max|x-x_i|<h时，展开的近似误差较小，然而这要求h比较大
当h较小，使用更多的展开项（m比较大）

这个方法实用价值不大，因为密度估计有误差，而从分类的角度，有很多分类器可以代替。但是思路值得借鉴。

总结

- 非参数法的基本思想
  - 没有给定概率密度函数形式
  - 基于概率和密度的原始定义，以训练样本的局部分布近似**x**的局部密度
- Parzen window
- K-nearest neighbor (k-NN)
  - 1-nearest neighbr (1-NN), Error bound
  - 快速搜索
- 距离度量
  - Tangent distance
- Series expansion

下次课
（向世明老师）