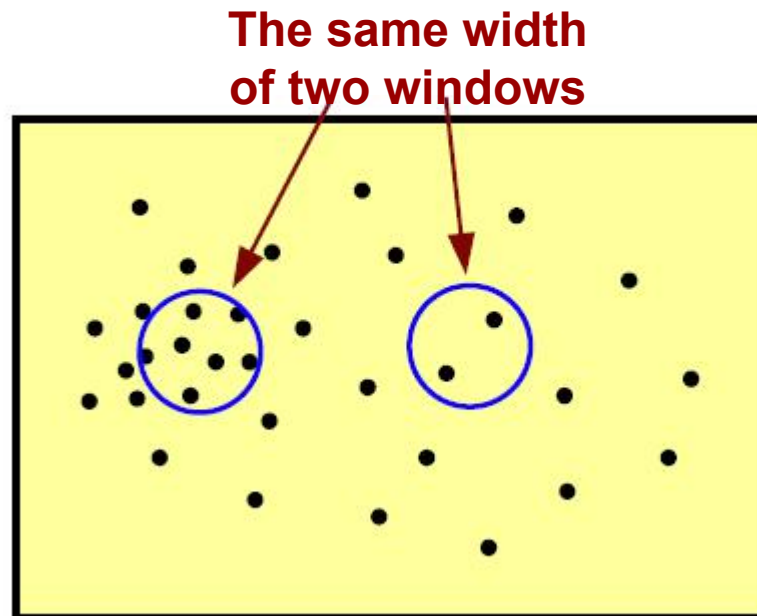


Ch 05. Non-parametric method

Part 2 k_n -Nearest neighbor estimation

The problem of Parzen Window Estimation

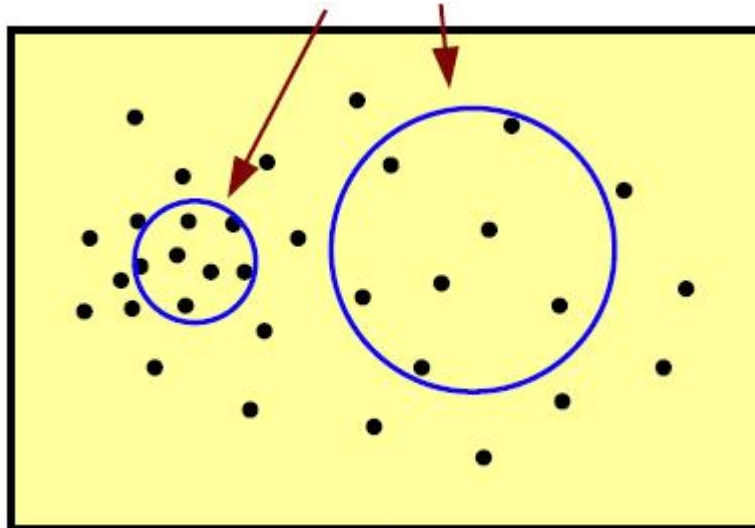
- If the distribution of $p(x)$ is non-uniform, using the same width of window in the entire feature space may not get satisfactory results.



k_n -Nearest neighbor estimation

- A method to solve the problem of Parzen window estimation with fixed window width
 - The window width is not fixed, but the number of samples k around x is fixed.
 - k is denoted as k_n , because it usually depends on the total number of samples n
 - The density around x is large, the window width becomes smaller (high resolution)
 - The density around x is small, the window width becomes larger (low resolution)
 - The k_n samples included by the window are called **the k_n nearest neighbors of x**

**The same number of samples
in the window**



k_n -Nearest neighbor estimation

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

The necessary and sufficient conditions for $p_n(\mathbf{x})$ to converge to the true distribution $p(\mathbf{x})$ are

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

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

- A common choice that satisfies this condition

$$k_n = \sqrt{n}$$

An Example

- One-dimensional distribution , $k_n = \sqrt{n}$

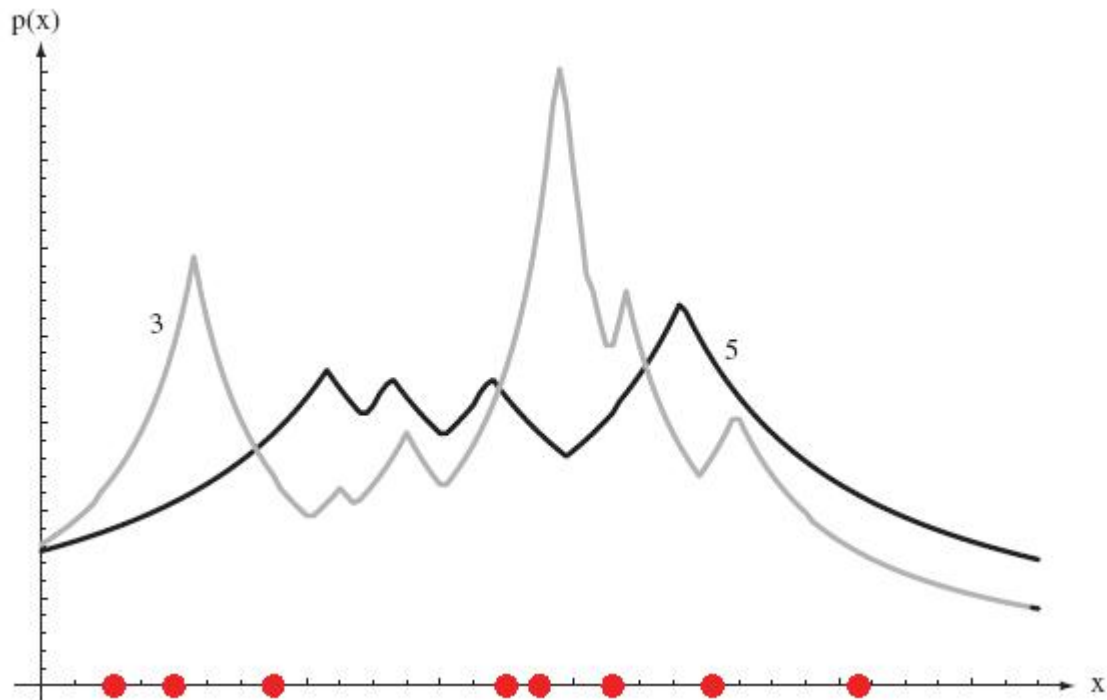
- When $n=1$,

$$p_n(x) = \frac{1}{2|x - x_1|}$$

- When $n > 1$,

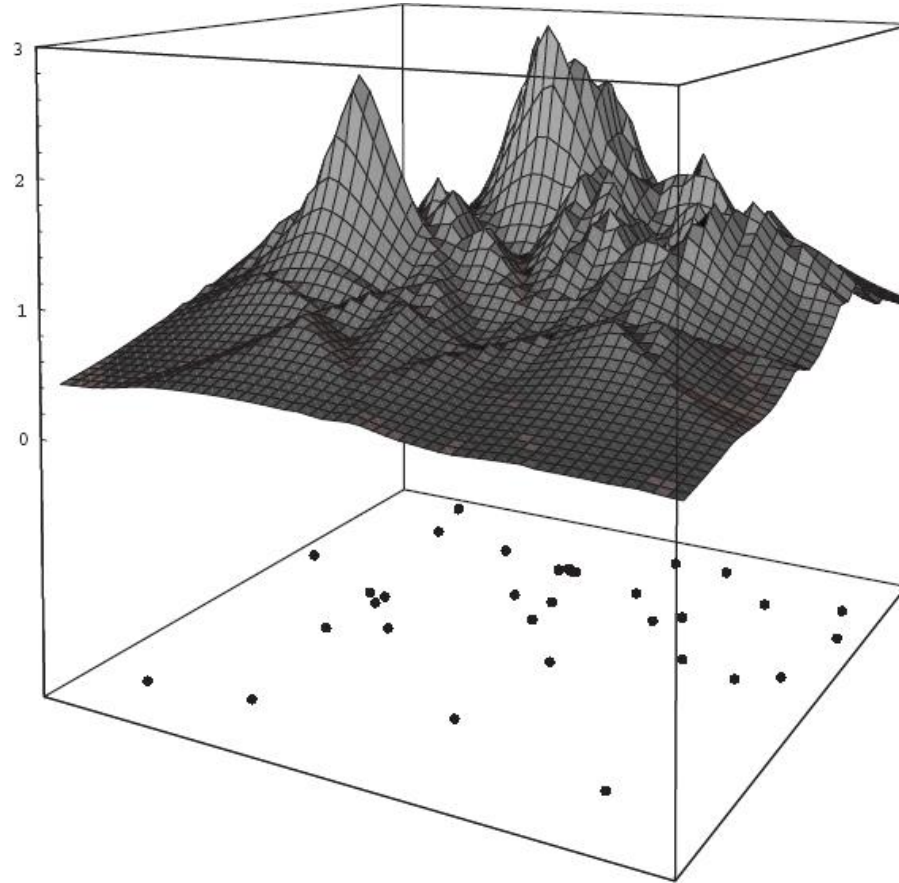
$$p_n(x) = \frac{1}{2\sqrt{n} \max_{i \in k_n \text{ 近邻}} (|x - x_i|)}$$

An Example



$n=8, k=3 \text{ or } 5$

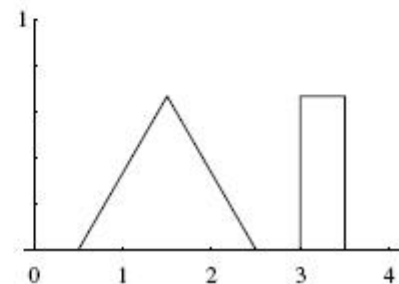
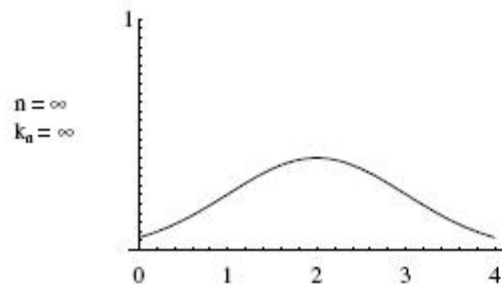
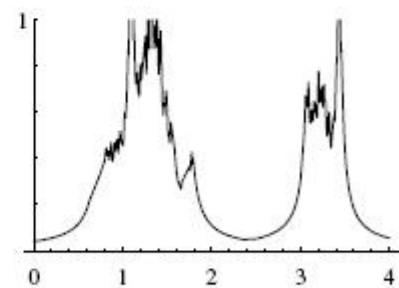
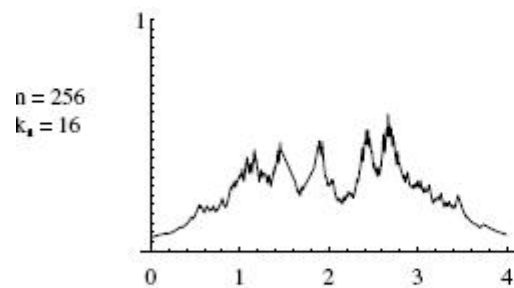
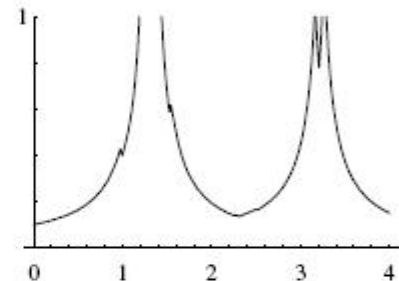
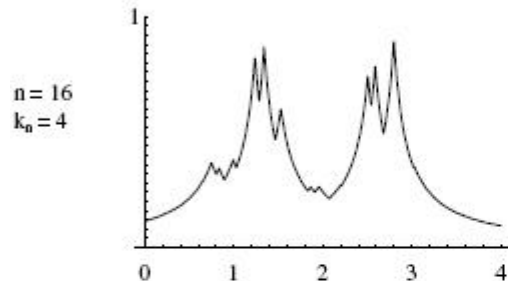
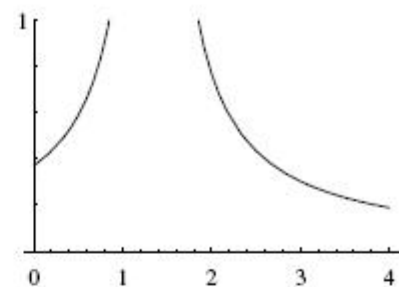
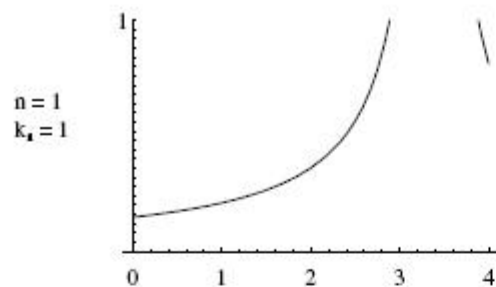
An Example



$K=5$

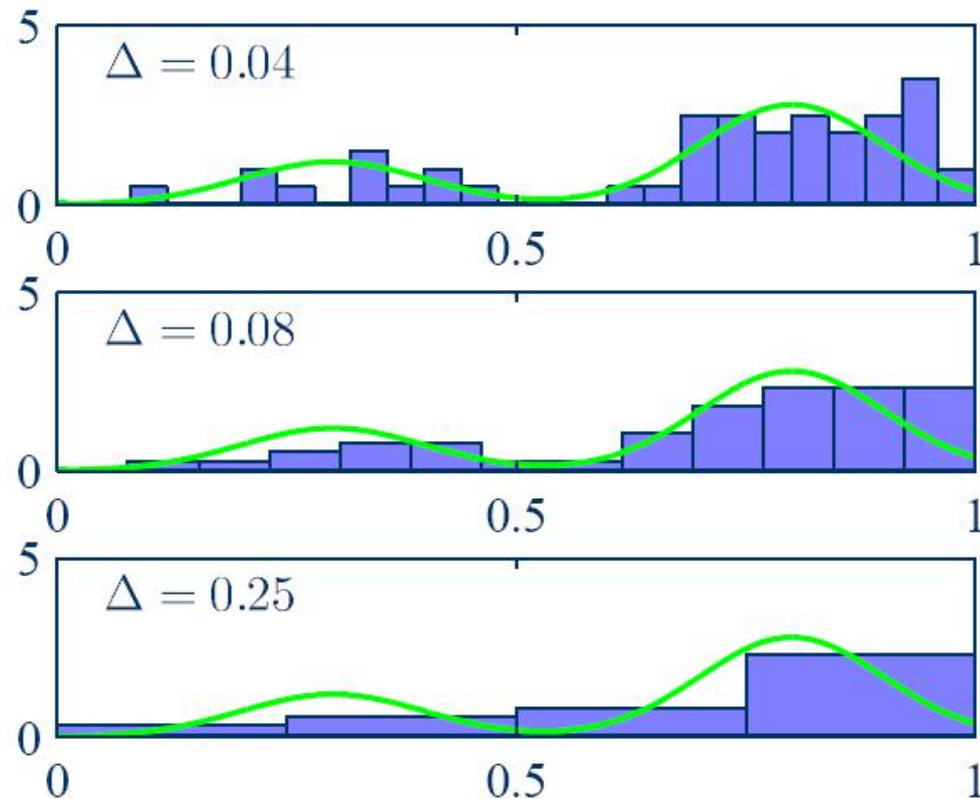
An Example

$$k_n = \sqrt{n}$$



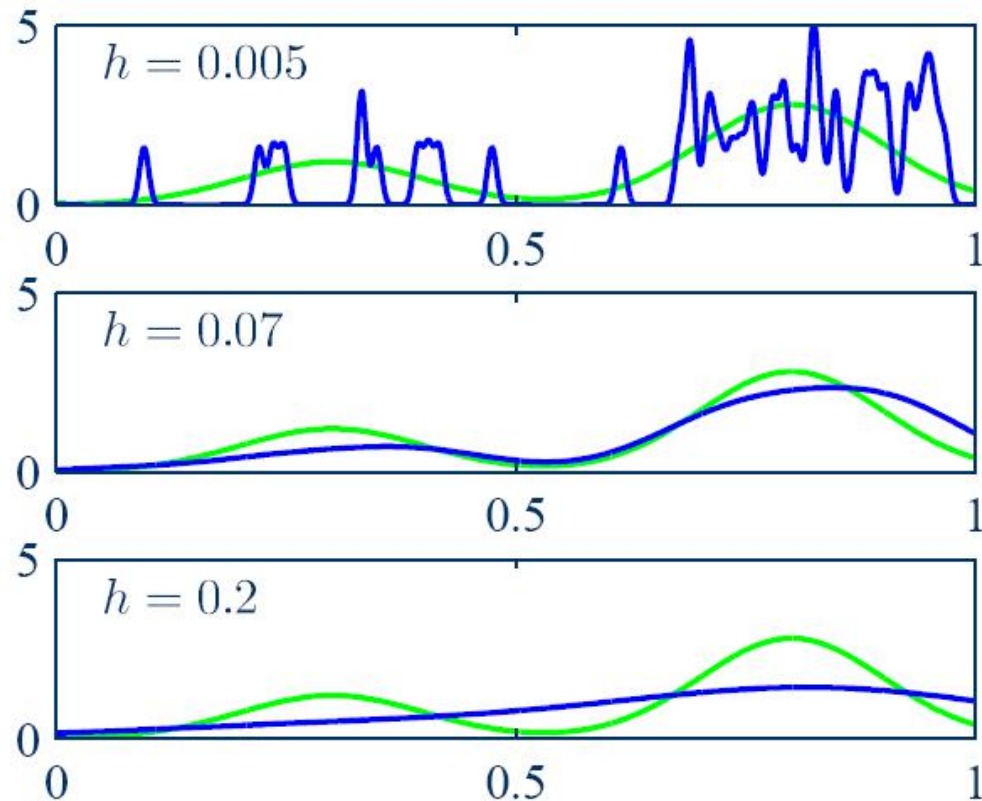
More Examples for Non-parametric Estimation

- Histogram estimation



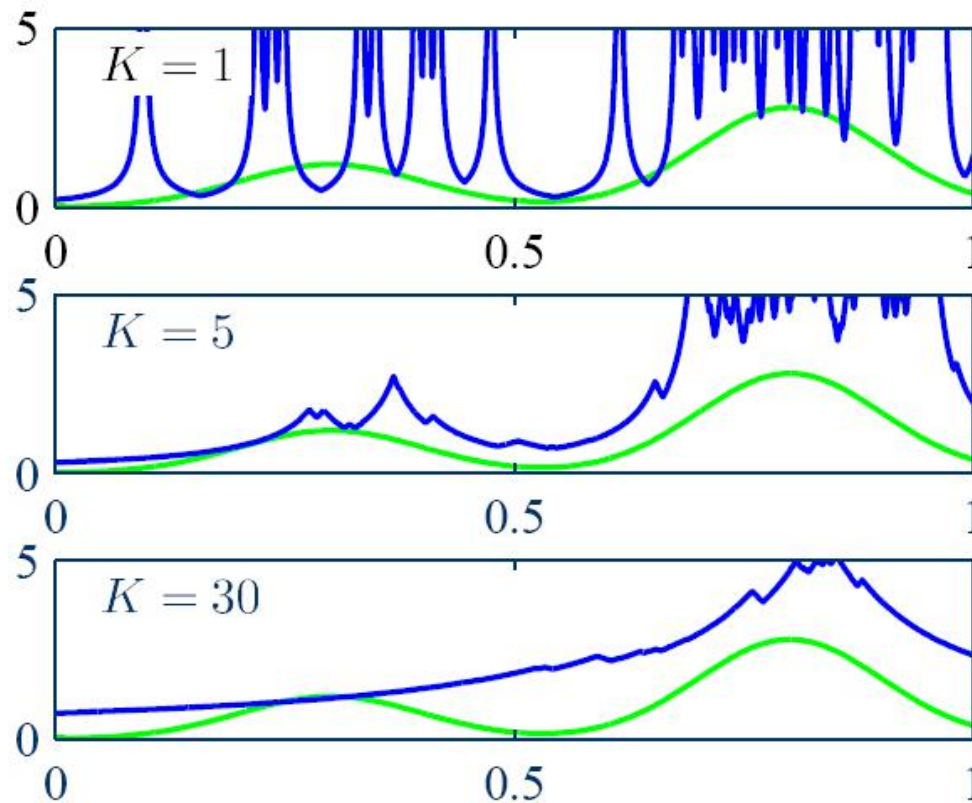
More Examples for Non-parametric Estimation

- Parzen window estimation

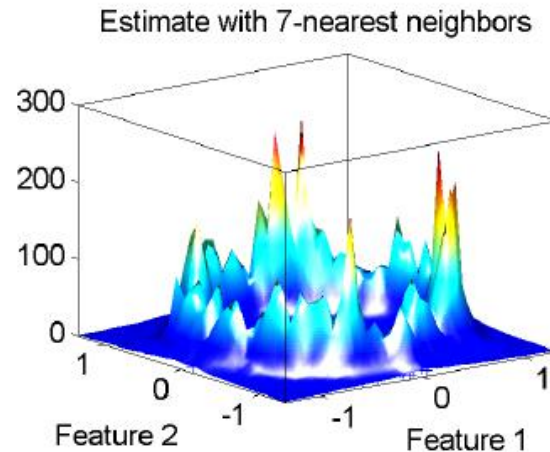
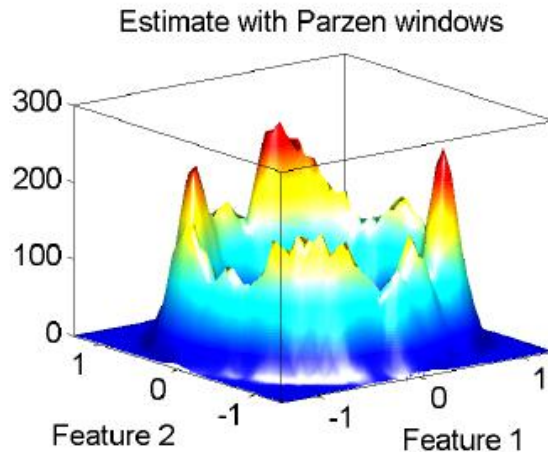
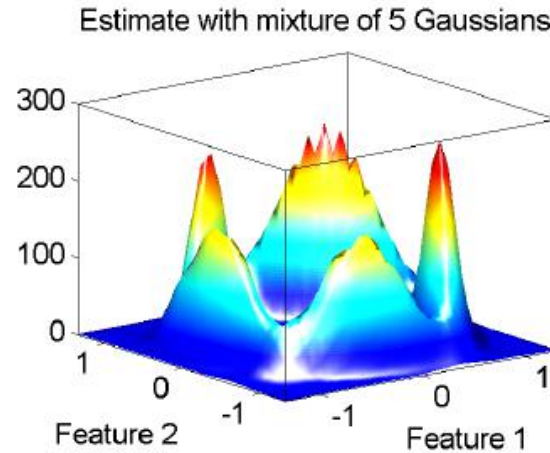
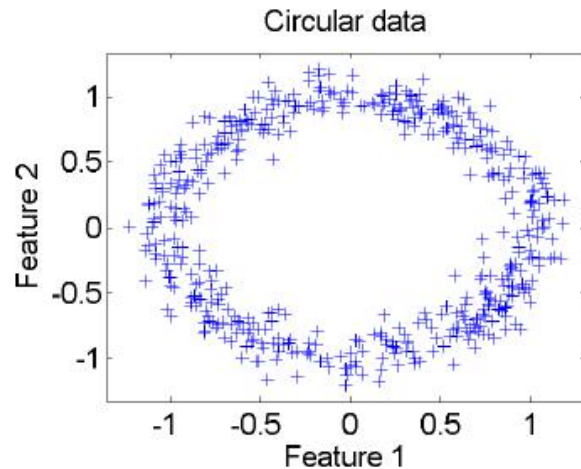


More Examples for Non-parametric Estimation

- k_n -Nearest neighbor estimation

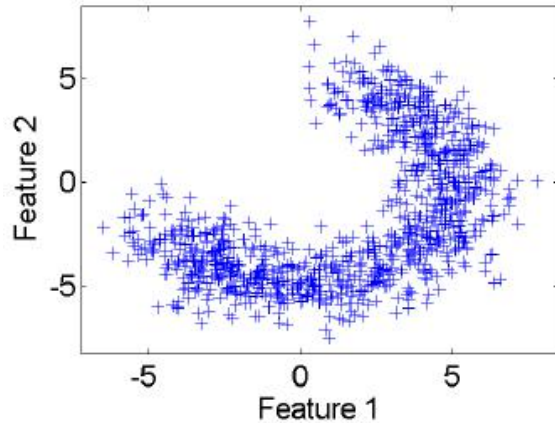


More Examples for Non-parametric Estimation

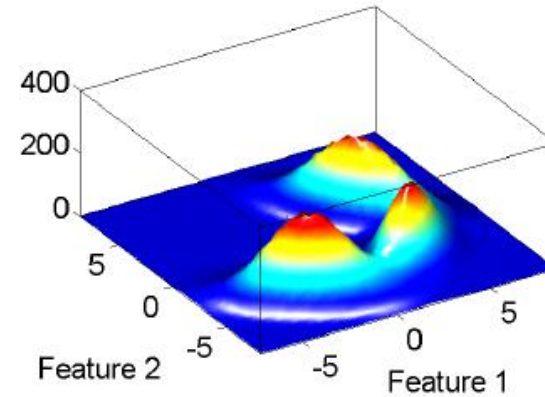


More Examples for Non-parametric Estimation

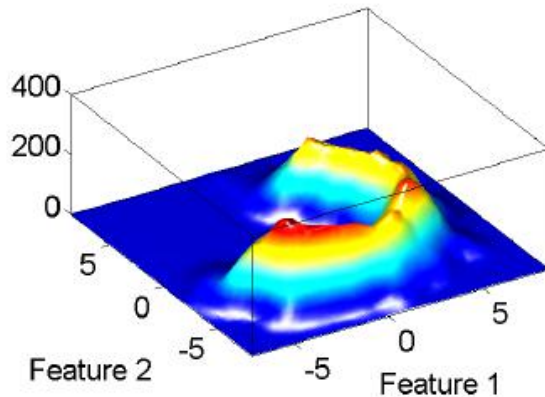
Banana shaped data



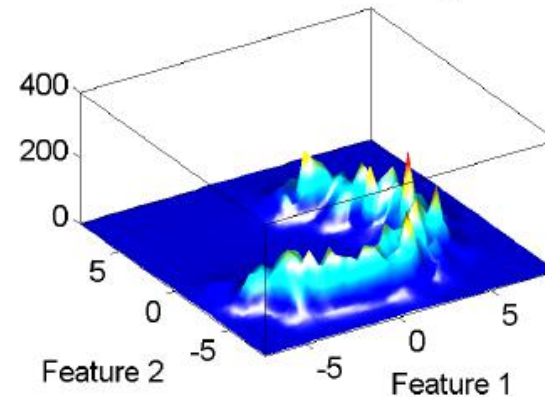
Estimate with mixture of 3 Gaussians



Estimate with Parzen windows



Estimate with 7-nearest neighbors



Ch 05. Non-parametric method

Part 3 k-nearest neighbor rule

Approaches to Pattern Classification

- **Approach 1:** Estimate class-conditional probability density $p(\mathbf{x} | \omega_i)$
 - Through $p(\mathbf{x} | \omega_i)$ and $P(\omega_i)$, calculate posterior probability $P(\omega_i | \mathbf{x})$ with Bayes' rule, then make decisions with maximum posterior probability
 - Two Methods
 - **Method 1a:** Parameter estimation of probability density
Based on parametric description of $p(\mathbf{x} | \omega_i)$
 - **Method 1b:** Non-Parametric estimation of probability density
Based on non-parametric description of $p(\mathbf{x} | \omega_i)$
- **Approach 2:** Estimate posterior probability $P(\omega_i | \mathbf{x})$
 - Don't have to estimate $p(\mathbf{x} | \omega_i)$ in advance
- **Approach 3:** Compute discrimination function
 - Don't have to estimate $p(\mathbf{x} | \omega_i)$ or $P(\omega_i | \mathbf{x})$

Non-parametric Estimation of Posterior Probability


- Suppose a region R near \mathbf{x} can be included into k samples, where k_i samples belong to the category ω_i ,

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

- Posterior probability

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

- Decision

- Parzen window estimation: select the category ω_i with the largest value of k_i / k
- k_n -Nearest neighbor estimation: select the category ω_i with the largest value of k_i  **k-nearest neighbor classifier**

Nearest Neighbor Rule

- k-nearest neighbor decision when **k=1**
 - Judge x as the category of the training sample x' closest to it
- Given training set $D = \{x_1, x_2, \dots, x_n\}$, which includes n samples from c different categories
- For test sample x , if $x_k \in D$ is the training sample closest to x (according to some distance measurement), then the nearest neighbor (1-NN) rule is

If x_k belongs to a class ω_j , the class of x is judged to be ω_j

- The nearest neighbor rule is a **suboptimal** method, and the usual error rate is greater than the least possible (i.e., Bayesian error rate)

Nearest Neighbor Rule

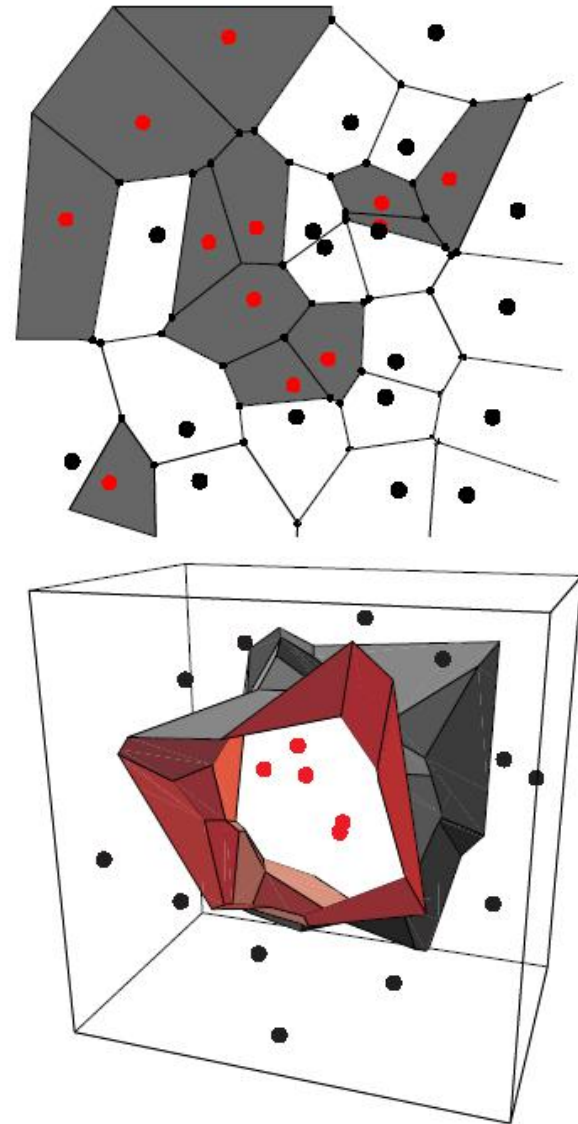
- Intuitive understanding
 - When the number of samples is large, \mathbf{x}' can be considered close enough to \mathbf{x} to make

$$P(\omega_i | \mathbf{x}') \approx P(\omega_i | \mathbf{x})$$

i.e. the nearest neighbor rule is a valid approximation to the true posterior probability

Voronoi Grid

- The nearest neighbor rule divides the feature space into a grid cell structure called Voronoi grid
 - Each unit contains a training sample point x'
 - The distance from any point x in the unit to x' is less than the distance from other training sample points
 - All sample points in this unit are judged to belong to the category of x'



Error Rate of The Nearest Neighbor Rule

- Given a training set $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, which includes n samples from c different categories
- For test sample \mathbf{x} , let $\mathbf{x}_k \in D$ be the training sample closest to \mathbf{x}
- The category labels for \mathbf{x} and \mathbf{x}_k are θ and φ , respectively
- **Conditional error probability**

$$e_{1NN}(\mathbf{x}, \mathbf{x}_k) = P(\theta \neq \varphi \mid \mathbf{x}, \mathbf{x}_k)$$

$$= \sum_{i=1}^c P(\theta = \omega_i, \varphi \neq \omega_i \mid \mathbf{x}, \mathbf{x}_k)$$

$$= \sum_{i=1}^c P(\theta = \omega_i \mid \mathbf{x}) P(\varphi \neq \omega_i \mid \mathbf{x}_k)$$

$$= \sum_{i=1}^c P(\theta = \omega_i \mid \mathbf{x}) [1 - P(\varphi = \omega_i \mid \mathbf{x}_k)]$$

Error Rate of The Nearest Neighbor Rule

- **Conditional error probability** (cont')

- When $n \rightarrow \infty$, suppose D contains enough samples to make $\lim_{n \rightarrow \infty} P(\phi = \omega_i | \mathbf{x}_k) = P(\theta = \omega_i | \mathbf{x})$

and when $n \rightarrow \infty$, then

$$e_{1NN}(\mathbf{x}, \mathbf{x}_k) \rightarrow \sum_{i=1}^c P(\omega_i | \mathbf{x}) - \sum_{i=1}^c [P(\omega_i | \mathbf{x})]^2 = 1 - \sum_{i=1}^c [P(\omega_i | \mathbf{x})]^2$$

- **Average error rate** (when $n \rightarrow \infty$)

$$\begin{aligned} e_{1NN} &= \int e_{1NN}(\mathbf{x}, \mathbf{x}_k) p(\mathbf{x}) d\mathbf{x} \\ &= \int \left\{ 1 - \sum_{i=1}^c [P(\omega_i | \mathbf{x})]^2 \right\} p(\mathbf{x}) d\mathbf{x} \end{aligned}$$

Error Bound of The Nearest Neighbor Rule

- The **lower bound** of the average error rate e_{1NN}

$$e_{1NN} \geq \text{Bayes error } P^*$$

- The **upper bound** of the average error rate e_{1NN}

- When $\sum_{i=1}^c [P(\omega_i | \mathbf{x})]^2$ takes the minimum of each \mathbf{x} , e_{1NN} is the maximum

- Let the true class of \mathbf{x} be ω_m , then the Bayesian error rate is expressed as

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

Error Bound of The Nearest Neighbor Rule

- The **upper bound** of the average error rate e_{1NN} (cont')
 - Given P^* (i.e. given $P(\omega_m | \mathbf{x})$)

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

This equation is smallest when the second term is the smallest. The second term is the smallest when

$P(\omega_i | \mathbf{x})$ is equal for all i except m , i.e.

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

Error Bound of The Nearest Neighbor Rule

- The **upper bound** of the average error rate e_{1NN} (cont')

- SO

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

or

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

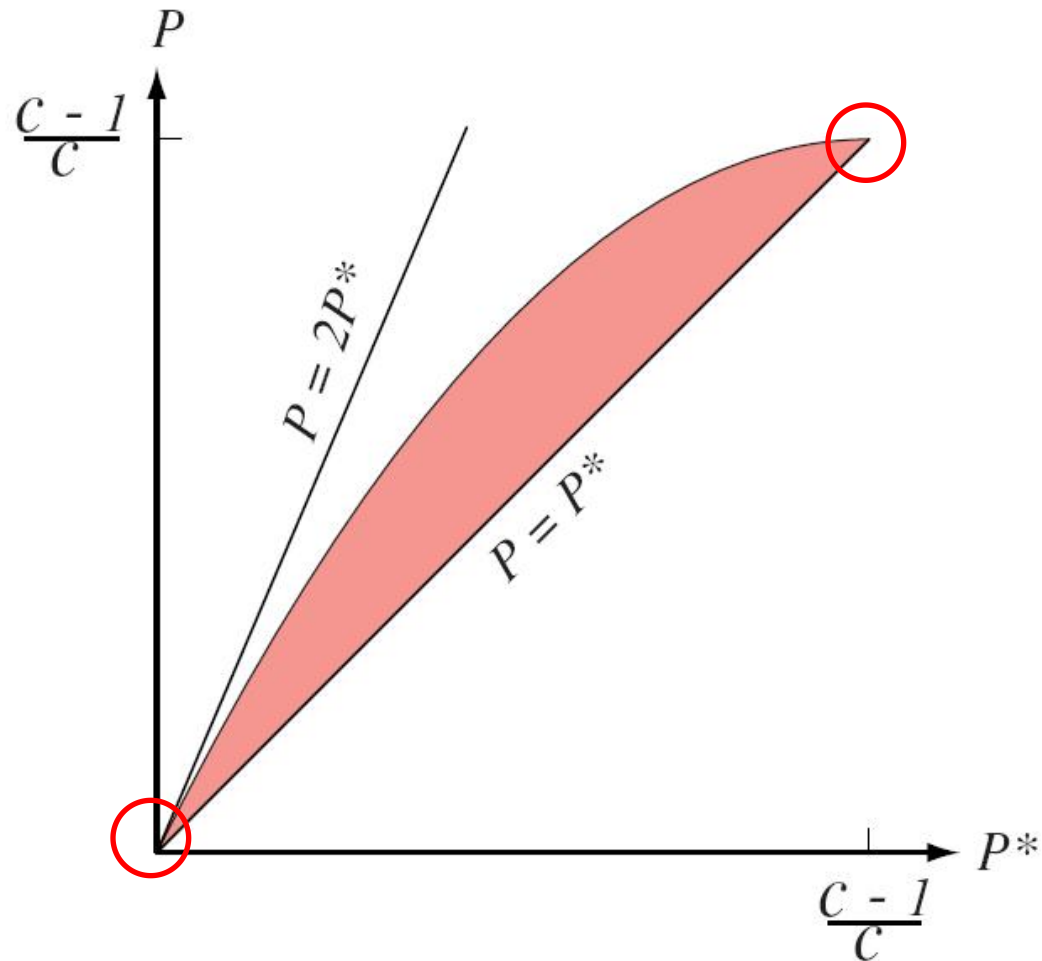
- SO

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

- When P^* is small, the upper bound of the nearest neighbor rule's average error rate is:

$$e_{1NN} \approx 2 P^*$$

Error Bound of The Nearest Neighbor Rule

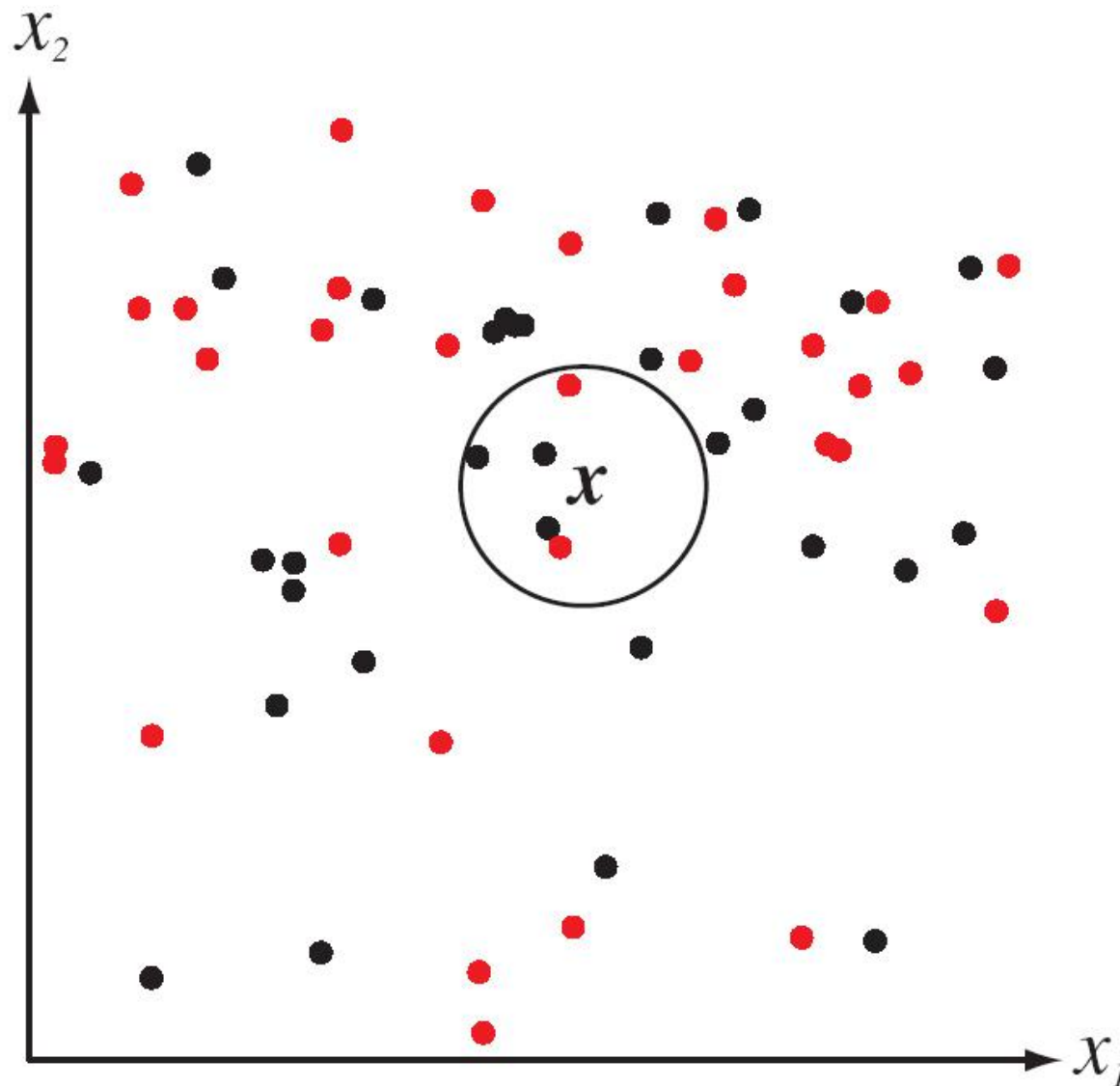


k-Nearest Neighbor Rule

- The **k-nearest neighbor (k-NN)** rule is an extension of the nearest neighbor (1-NN) rule, i.e. multiple nearest neighbors are considered
- Given a training set $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, which includes n samples from c different categories
- For test sample x , let the set $S \subset D$ contain k training samples closest to x
- k-nearest neighbor rule

If ω_j is the class that appears most frequently in S , then the class of x is judged to be ω_j

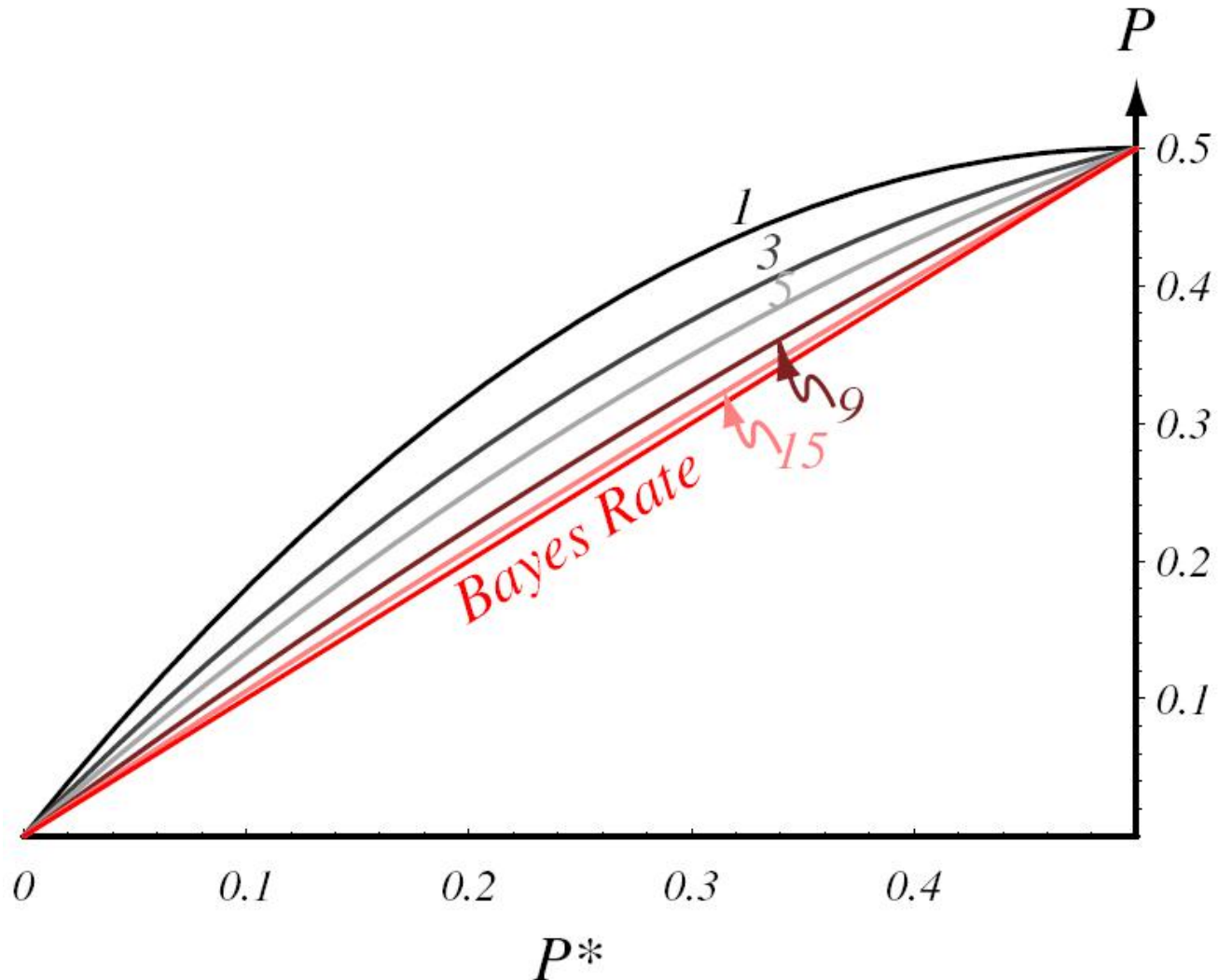
k-Nearest Neighbor Rule



k-Nearest Neighbor Rule

- The **lower bound** of the average error rate e_{kNN}
Bayes error rate P^*
- The **upper bound** of the average error rate e_{kNN}
 - When $n \rightarrow \infty$, and $k \rightarrow \infty$, then $e_{kNN} \rightarrow P^*$
 - When k is large enough, but small enough relative to n , applying the k -NN rule to the large number of samples approximates the optimal decision

Error Bound of The k-Nearest Neighbor Rule



The Choice of k

- The k-nearest neighbor rule can be regarded as a way to estimate the posterior probability $P(\omega_i|\mathbf{x})$ directly from the sample
- To get a reliable estimate (with a low margin of error), bigger k is better
- In order for $P(\omega_i|\mathbf{x}')$ to be as close to $P(\omega_i|\mathbf{x})$ as possible, the nearest neighbors of \mathbf{x} are as close to \mathbf{x} as possible, i.e. k is as small as possible
- According to the actual problem, the value of k should be trade-off
- When n tends to infinity, and k also tends to infinity at a slower speed, the k-nearest neighbor rule is the optimal classification rule

Example

$k = 3$ (Odd number), $x = (0.10, 0.25)$

Training set	category
(0.15, 0.35)	ω_1
(0.10, 0.28)	ω_2
(0.09, 0.30)	ω_5
(0.12, 0.20)	ω_2

k nearest neighbors of x:

$$\{(0.10, 0.28, \omega_2); (0.09, 0.30, \omega_5); (0.12, 0.20, \omega_2)\}$$

According to the k-nearest neighbor rule, the category of x is judged to be ω_2

Computational Complexity

- Direct method
 - Suppose training set D consists of n d -dimensional samples
 - Given a test sample x , the distance between it and all sample x_i in the training set should be calculated, and the computational complexity is $O(dn)$.
 - When n is large, the time and space complexity will be high!
- The method to reduce computational complexity
 - Calculate partial distance
 - Pre-build structure
 - Clip training samples

Calculate Partial Distance

- When calculating distances, only one subset \mathbf{r} of d dimensions is used

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

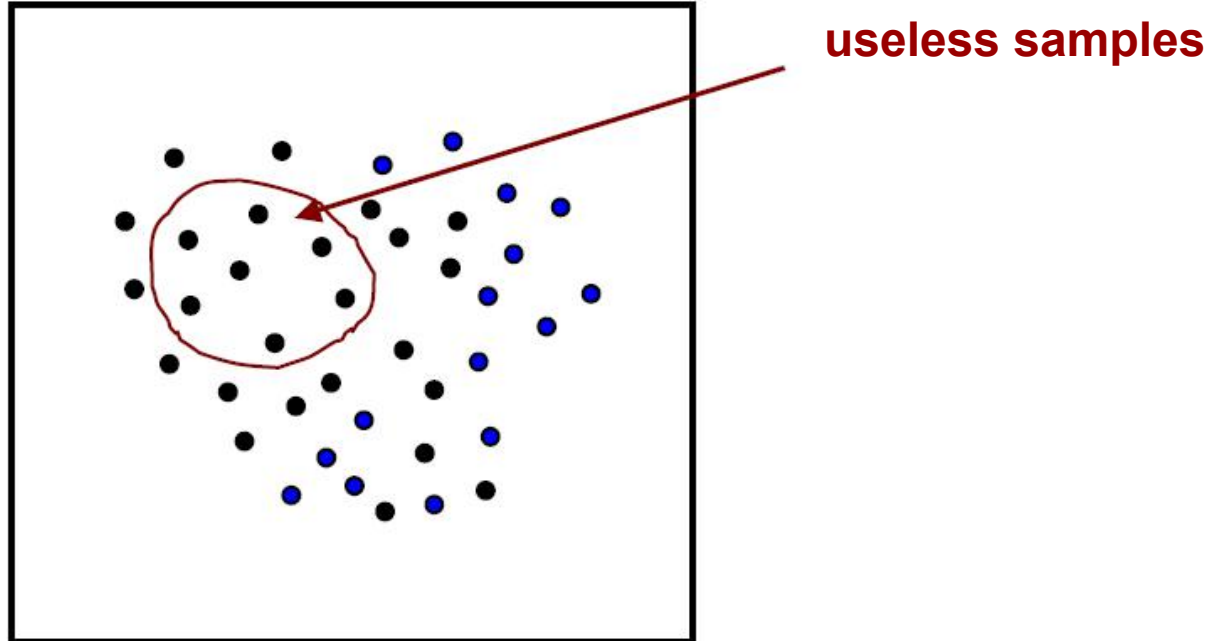
- When more dimensions are added gradually, the value of partial distance is strictly non-decreasing
- How to save computational cost when calculating the nearest neighbor of test sample \mathbf{x} ?
 - When calculating the nearest neighbor of \mathbf{x} , the current nearest neighbor of \mathbf{x} can be updated with each training sample examined
 - If the partial distance from \mathbf{x} to a certain training sample on subset \mathbf{r} is already greater than the distance from its nearest neighbor, the calculation can be stopped immediately, discarding the training sample and continuing to investigate the next sample
 - This technique is especially useful when calculating distances if the dimensions with large variances are calculated first

Pre-build Structure

- Pre-build some form of search tree, and organize them according to the relative distance between training sample points
- After the search tree is established, searching the nearest neighbor to x only need to visit a portion of the entire tree, thus the amount of computational cost can be saved
- For example
 - Suppose that the samples obey a uniform distribution within the unit square $U\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right)$
 - Select four root nodes $\begin{pmatrix} 1/4 \\ 1/4 \end{pmatrix}$, $\begin{pmatrix} 1/4 \\ 3/4 \end{pmatrix}$, $\begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}$ and $\begin{pmatrix} 3/4 \\ 3/4 \end{pmatrix}$
 - For test point x , first calculate the distance from it to the 4 root nodes, and select the nearest one. Then, the search is only limited to the quadrant represented by this root node, and the remaining 3/4 training samples are not necessary to access
 - There's no guarantee of finding x 's true nearest neighbor

Training Samples Clip

- Eliminate the "useless" training samples
- Which samples are "useless"?
 - Samples surrounded by samples of the same category!



Training Samples Clip

- Nearest-neighbor editing

```
1 begin initialize  $j = 0$ ,  $\mathcal{D}$  = data set,  $n$  = #prototypes
2       construct the full Voronoi diagram of  $\mathcal{D}$ 
3       do  $j \leftarrow j + 1$ ; for each prototype  $\mathbf{x}'_j$ 
4           Find the Voronoi neighbors of  $\mathbf{x}'_j$ 
5           if any neighbor is not from the same class as  $\mathbf{x}'_j$  then mark  $\mathbf{x}'_j$ 
6       until  $j = n$ 
7   Discard all points that are not marked
8   Construct the Voronoi diagram of the remaining (marked) prototypes
9 end
```

Ch 05. Non-parametric Method

Part 4 Distance Metric

Distance Metric

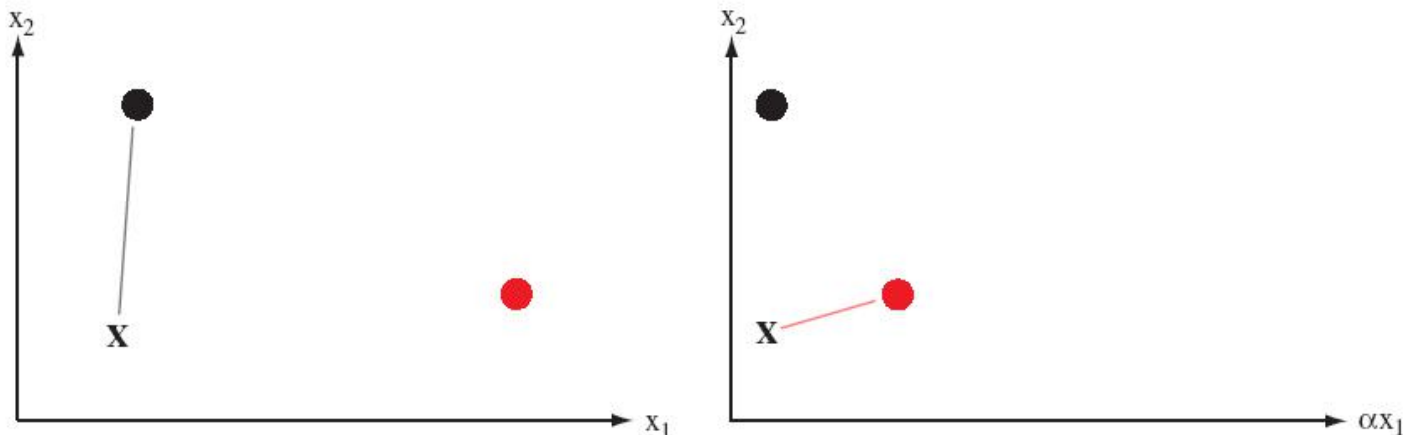
- The nearest neighbor rule or k-nearest neighbor rule is based on a metric that measures the distance between patterns (samples)
- Distance metric is one of the core issues in the field of pattern recognition
- A generic representation of metric $D(\mathbf{a}, \mathbf{b})$
- The properties that metric must satisfy
 - **Nonnegativity:** $D(\mathbf{a}, \mathbf{b}) \geq 0$
 - **Reflexivity:** $D(\mathbf{a}, \mathbf{b}) = 0$ if and only if $\mathbf{a} = \mathbf{b}$
 - **Symmetry:** $D(\mathbf{a}, \mathbf{b}) = D(\mathbf{b}, \mathbf{a})$
 - **Triangle inequality:** $D(\mathbf{a}, \mathbf{b}) + D(\mathbf{b}, \mathbf{c}) \geq D(\mathbf{a}, \mathbf{c})$

Euclidean Distance

- **Euclidean distance** in d-dimensional space

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

- The transformation of characteristic scale will seriously affect the nearest neighbor relationship calculated by Euclidean distance



Euclidean Distance

- Solution
 - Perform scale equalization on the distribution of each dimension (feature), so that the range of change on each dimension is equal, such as all normalization into interval $[0, 1]$

Minkowski Distance

- **Minkowski distance** in d-dimensional space

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

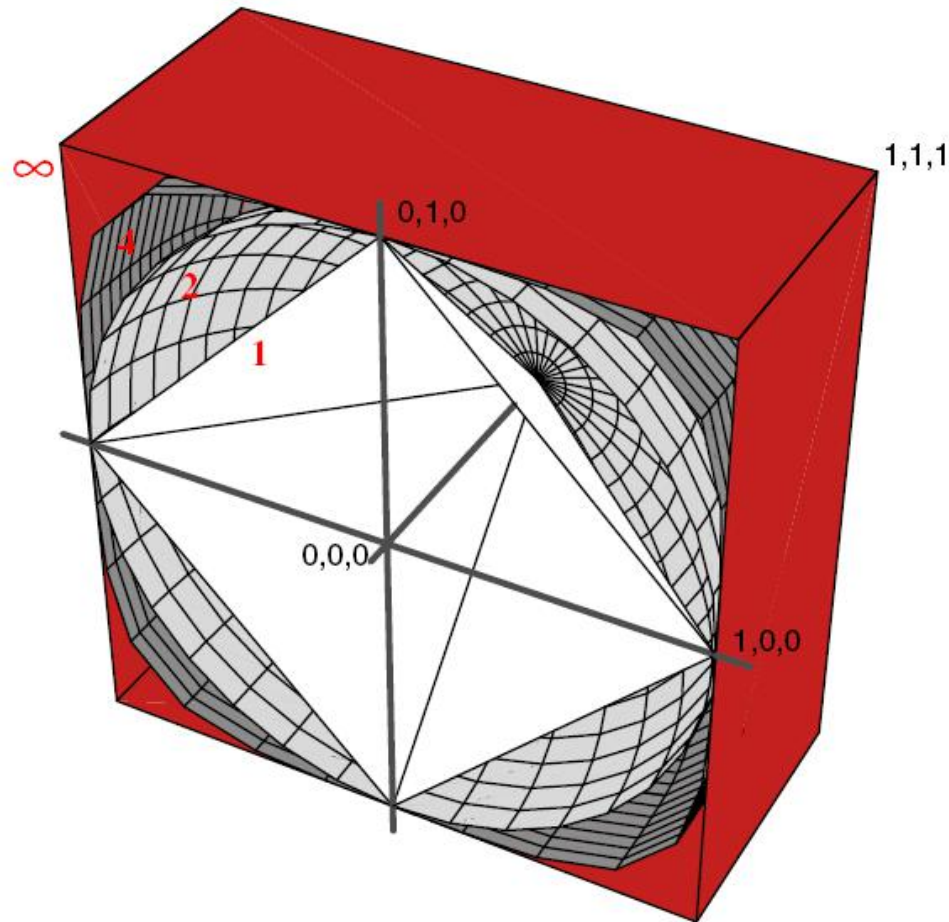
- Also known as the L_k -norm
- L_2 -norm——Euclidean distance
- L_1 -norm——Manhattan distance (block distance)

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

- L_∞ -norm——The maximum projection distance of a and b on d-axes

Minkowski Distance

- Isometric surface to the origin



Mahalanobis Distance

- Mahalanobis distance (马氏距离) considers the covariance Σ between features when calculating the distance

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

- The relationship between Mahalanobis distance and multivariable normal distribution

$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \Sigma)$$

$$\begin{aligned} p(\mathbf{x}) &= \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right] \\ &= \alpha \exp \left[-\frac{D_{Mahalanobis}^2(\mathbf{x}, \boldsymbol{\mu})}{2} \right] \end{aligned}$$

Mahalanobis Distance

- For example

- \mathbf{a} : $[0.8, 0.2]^T$, \mathbf{b} : $[0.1, 0.5]^T$ extracted from normal distribution $N(0, \Sigma)$,

where $\Sigma = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix}$, Calculate the Mahalanobis distance between \mathbf{a} and \mathbf{b}

- Solution: $D_{Mahalanobis}(\mathbf{a}, \mathbf{b}) = \sqrt{(\mathbf{a} - \mathbf{b})^T \Sigma^{-1} (\mathbf{a} - \mathbf{b})}$
$$= \sqrt{\left(\begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix} - \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix} \right)^T \begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix}^{-1} \left(\begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix} - \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix} \right)}$$
$$= \sqrt{\begin{bmatrix} 0.7 \\ -0.3 \end{bmatrix}^T \begin{bmatrix} 1/0.2 & 0 \\ 0 & 1/0.1 \end{bmatrix} \begin{bmatrix} 0.7 \\ -0.3 \end{bmatrix}}$$
$$= \sqrt{\frac{0.7^2}{0.2} + \frac{(-0.3)^2}{0.1}} = \sqrt{\frac{0.7^2}{0.2} + \frac{(-0.3)^2}{0.1}} = 1.83$$

Distance Metric Between Sets

- **Tanimoto distance**

$$D_{Tanimoto}(S_1, S_2) = \frac{n_1 + n_2 - 2n_{12}}{n_1 + n_2 - n_{12}}$$

- n_1 and n_2 are the numbers of elements in set S_1 and S_2 respectively
- n_{12} is the number of elements in the intersection of two sets
- Application scenarios
 - The two patterns (features) are either the same or different, and the similarity of a certain level cannot be calculated
 - For example, for the Tanimoto distance between two words, each word can be regarded as a set of letters

Distance Metric Between Sets

- **Tanimoto distance**

- For example

According to the Tanimoto distance, determine which of the following words most closely resembles 'pat':

'cat', 'pots', 'pattern'

- Solution

$$D_{Tanimoto}('pat', 'cat') = \frac{3 + 3 - 2 \times 2}{3 + 3 - 2} = 0.5$$

$$D_{Tanimoto}('pat', 'pots') = \frac{3 + 4 - 2 \times 2}{3 + 4 - 2} = 0.6$$

$$D_{Tanimoto}('pat', 'pattern') = \frac{3 + 7 - 2 \times 3}{3 + 7 - 3} = 0.57$$

So 'cat' is the closest word to 'pat'

Distance Metric Between Sets

- **Hausdorff distance**

- "The maximum value of the minimum distance from a point in one set to a point in another set"

$$D_{\text{Hausdorff}}(A, B) = \max \left(\max_{a \in A} \min_{b \in B} d(a, b), \max_{b \in B} \min_{a \in A} d(a, b) \right)$$

- $d(a, b)$ is a certain metric of the distance between two points a and b
 - Euclidean distance
 - Manhattan distance
 - Mahalanobis distance
 -

Distance Metric Between Sets

- **Hausdorff distance**

- For example

Compute the Hausdorff distance between

$$A = \left\{ \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}, \begin{bmatrix} 0.3 \\ 0.8 \end{bmatrix} \right\} \quad \text{and} \quad B = \left\{ \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \right\}$$

- Solution

$$d(a_1, b_1) = 0.5$$

$$d(a_1, b_2) = 0.61$$

$$d(a_2, b_1) = 0.36$$

$$d(a_2, b_2) = 0.64$$

$$\begin{aligned} D_{\text{Hausdorff}}(A, B) &= \max \left(\max_{a \in A} \min_{b \in B} d(a, b), \max_{b \in B} \min_{a \in A} d(a, b) \right) \\ &= \max(\max(0.5, 0.36), \max(0.36, 0.61)) \\ &= \max(0.5, 0.61) = 0.61 \end{aligned}$$

Distance Metric Between Sets

- **Hausdorff distance**

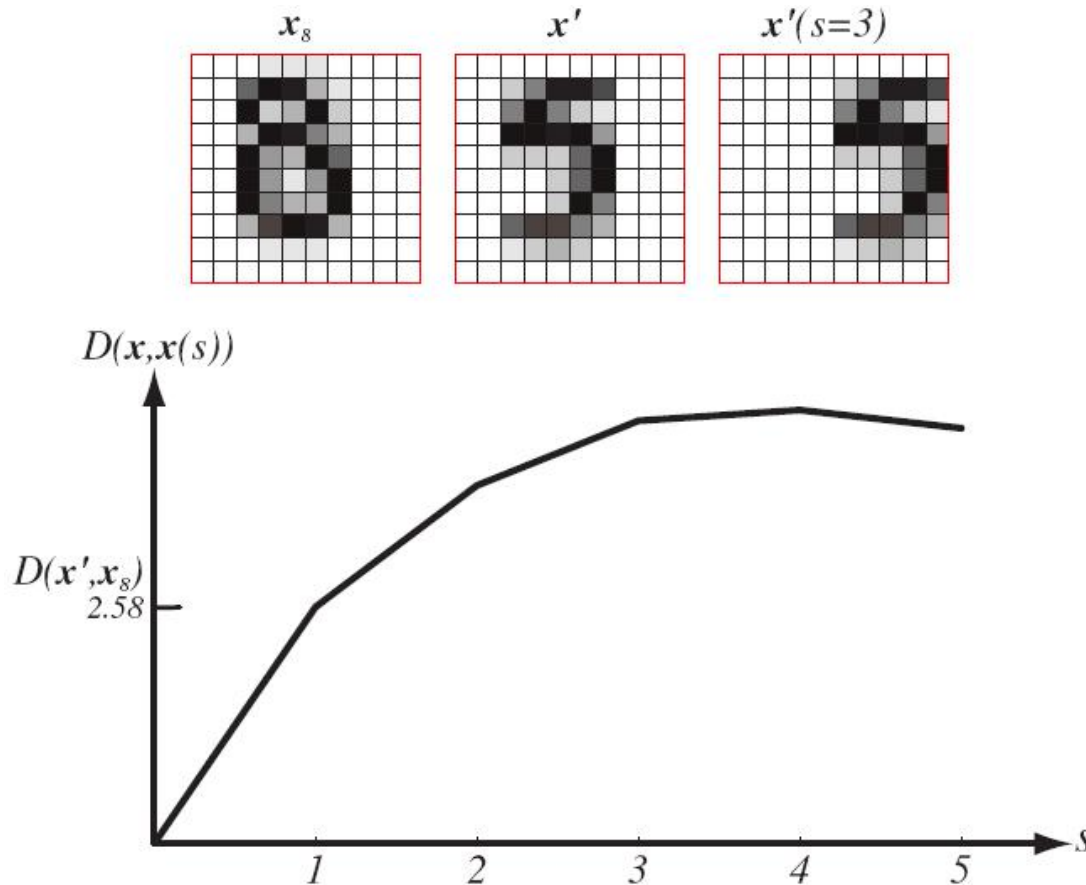
- Exercise

Compute the Hausdorff distance between

$$A = \left\{ \begin{bmatrix} -5 \\ 3 \end{bmatrix}, \begin{bmatrix} 9 \\ -2 \end{bmatrix} \right\} \quad \text{and} \quad B = \left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right\}$$

Tangent Space Distance

- In many practical problems, arbitrarily choosing a distance metric, such as the most commonly used Euclidean distance, may result in poor results



Tangent Space Distance

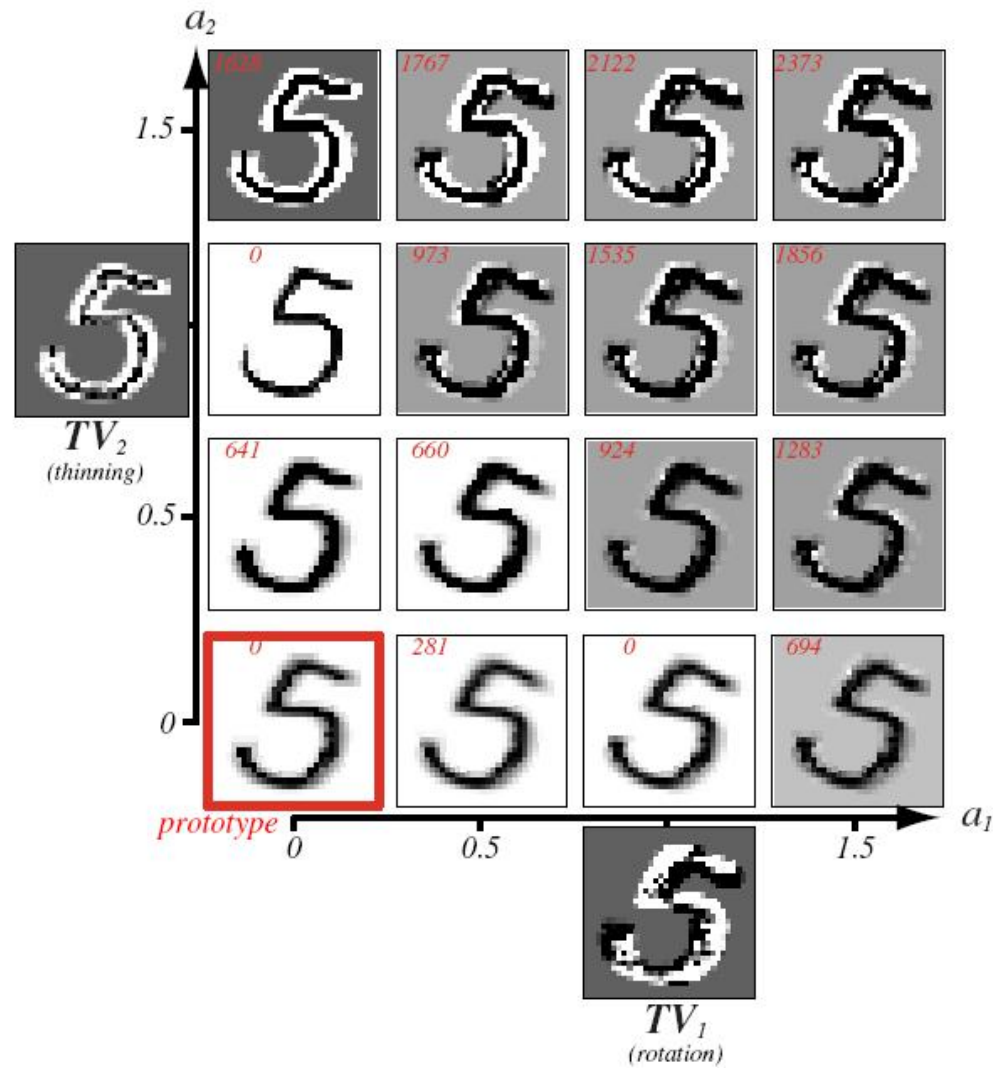
- invariance (不变量) problem
 - Translation
 - Rotation
 - Scale transformation
 - Thinning
 -
- Solution
 - Pretreatment
 - Use more general (with ** invariance) distance metric

Tangent Space Distance

- Tangent space distance has **the invariance of arbitrary transformation**
 - Suppose that the problem might involve r transformations
 - Perform all possible transformations for each training sample \mathbf{x}' , expressed as $\mathcal{F}_i(\mathbf{x}'; \alpha_i)$, $i=1,2,\dots,r$, where α_i is the parameter of the i -th transformation, such as translation distance, rotation angle, etc.
 - For each transformation, a tangent vector (切向量) can be constructed: $\mathbf{TV}_i = \mathcal{F}_i(\mathbf{x}'; \alpha_i) - \mathbf{x}'$
 - The tangent vectors of all transformations are spanned into a tangent space (切空间) of \mathbf{x}' , which is a linear approximation of all possible transformations of \mathbf{x}' , where each point corresponds to a possible transformation
 - The tangent space distance from test point \mathbf{x} to \mathbf{x}' is the minimum distance from \mathbf{x} to \mathbf{x}' , so it can be considered to have arbitrary transformation invariance

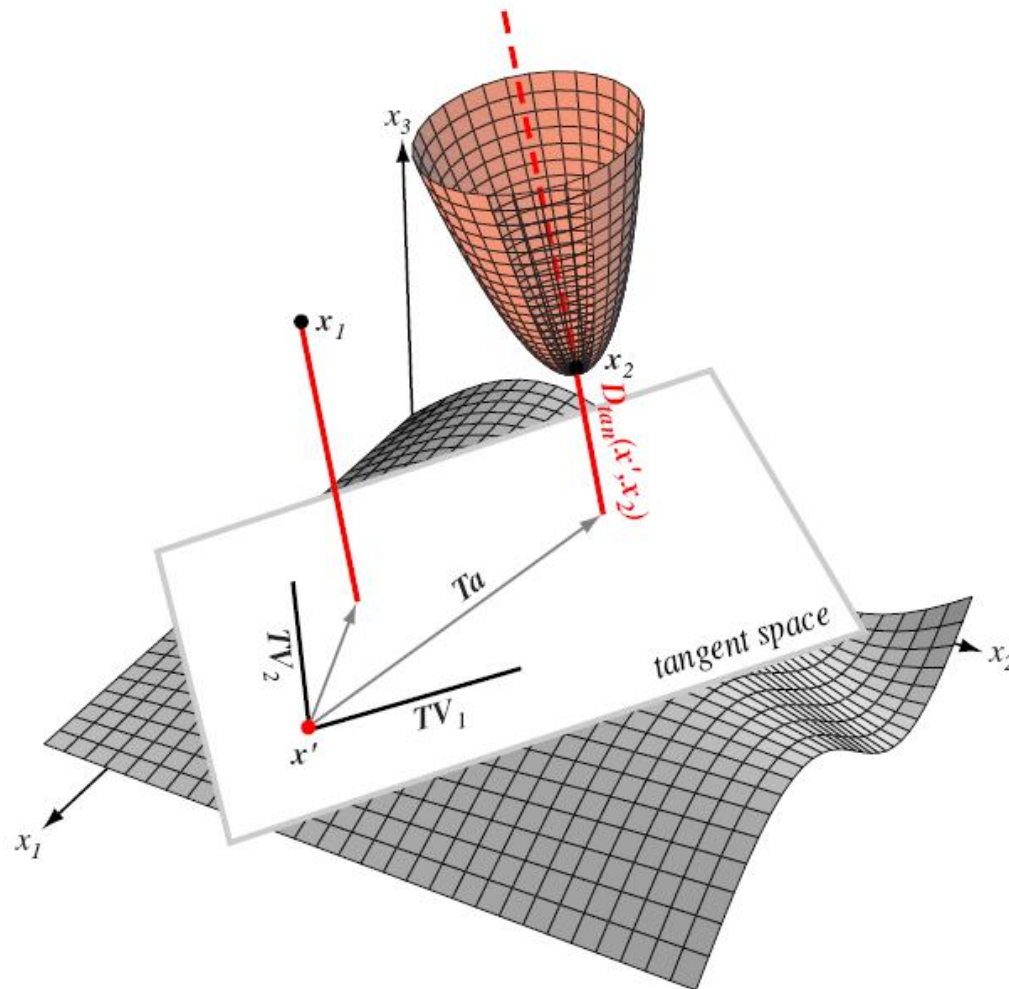
Tangent Space Distance

- For example: tangent space of rotation and thinning



Tangent Space Distance

- Calculate the tangent space distance from x to x'



Tangent Space Distance

- The nearest neighbor classifier based on tangent space distance usually has high accuracy
- However, the calculation of tangent space distance requires the designer to know all possible transformations in advance and be able to apply these transformations on each prototype point (training sample point), which sometimes cannot be satisfied in practice
- The computational complexity of tangent space distance is high, and the computational complexity may be unbearable when the data set is large