



Chapter 4

Non-Parameter Estimation

Contents

- Introduction
- Parzen Windows
- K-Nearest-Neighbor Estimation
- Classification Techniques
 - The Nearest-Neighbor rule(1-NN)
 - The Nearest-Neighbor rule(k-NN)
- Distance Metrics

Bayes Rule for Classification

MIMA

$$P(\omega_i | x) = \frac{p(x | \omega_i) p(\omega_i)}{p(x)}$$

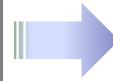
- To compute the posterior probability, we need to know the prior probability and the likelihood.
- Case I: $p(x | \omega_i)$ has certain parametric form
 - Maximum-Likelihood Estimation
 - Bayesian Parameter Estimation
- Problems:
 - The assumed parametric form may not fit the ground-truth density encountered in practice, e.g., assumed parametric form: **unimodal**; ground-truth: **multimodal**

Non-Parameter Estimation

MIMA

- Case II: $p(x | \omega_i)$ doesn't have parametric form
- How?

*Let the data
speak for
themselves!*



*Parzen Windows
 K_n -Nearest-Neighbor*

Goals

- Estimate class-conditional densities

$$p(\mathbf{x} | \omega_i)$$

- Estimate posterior probabilities

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

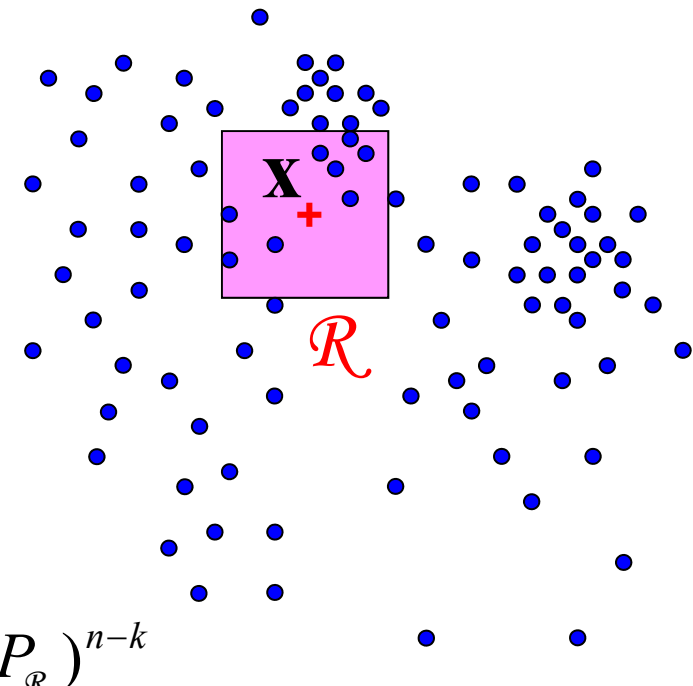
Density Estimation

- Assume $p(\mathbf{x})$ is continuous, and \mathcal{R} is small
- Fundamental fact
 - The probability of a vector \mathbf{x} fall into a region \mathcal{R} :

$$\begin{aligned} P(\mathbf{X} \in \mathcal{R}) &= \int_{\mathcal{R}} p(\mathbf{x}') d\mathbf{x}' \approx p(\mathbf{x}) \int_{\mathcal{R}} d\mathbf{x}' \\ &= p(\mathbf{x}) V_{\mathcal{R}} = P_{\mathcal{R}} \end{aligned}$$

Given n examples (i.i.d.) $\{x_1, x_2, \dots, x_n\}$, let \mathcal{K} denote the random variable representing number of samples falling into \mathcal{R} , \mathcal{K} will take Binomial distribution:

$$K \sim B(n, P_{\mathcal{R}}) \implies P(K = k) = \binom{n}{k} P_{\mathcal{R}}^k (1 - P_{\mathcal{R}})^{n-k}$$



n samples

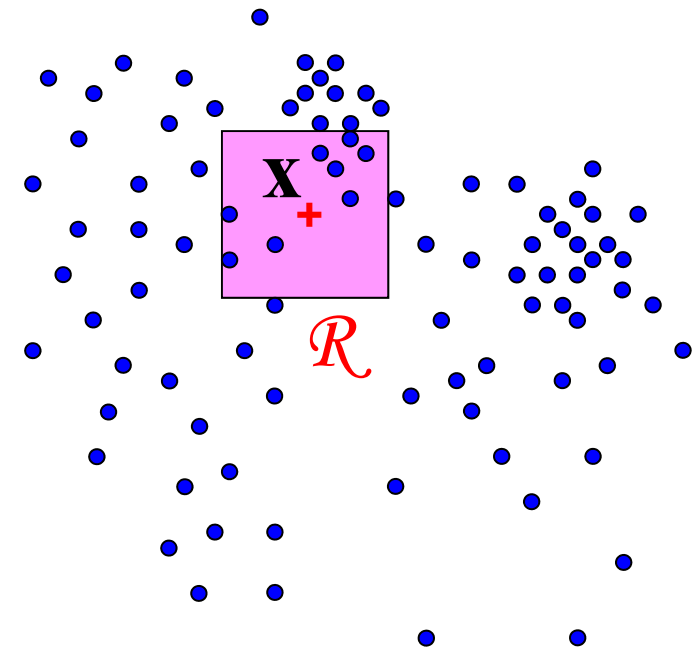
Density Estimation

- Assume $p(\mathbf{x})$ is continuous, and \mathcal{R} is small
- Fundamental fact
 - The probability of a vector \mathbf{x} fall into a region \mathcal{R} :

$$\begin{aligned} P(\mathbf{X} \in \mathcal{R}) &= \int_{\mathcal{R}} p(\mathbf{x}') d\mathbf{x}' \approx p(\mathbf{x}) \int_{\mathcal{R}} d\mathbf{x}' \\ &= p(\mathbf{x}) V_{\mathcal{R}} = P_{\mathcal{R}} \end{aligned}$$

$$\left. \begin{aligned} p(\mathbf{x}) V_{\mathcal{R}} &= P_{\mathcal{R}} \\ E[K] &= n P_{\mathcal{R}} \end{aligned} \right\} \Rightarrow p(\mathbf{x}) = \frac{E[K] / n}{V_{\mathcal{R}}}$$

Let $k_{\mathcal{R}}$ denote the actual number of samples in \mathcal{R} $\Rightarrow p(\mathbf{x}) \approx \frac{k_{\mathcal{R}} / n}{V_{\mathcal{R}}}$



Density Estimation

$$p(\mathbf{x}) \approx \frac{k_{\mathcal{R}} / n}{V_{\mathcal{R}}}$$

- Use subscript **n** to take sample size into account

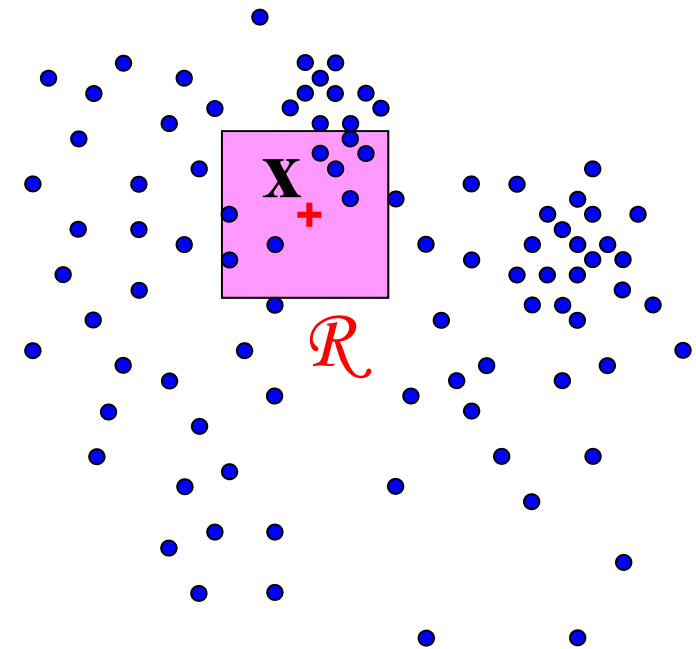
$$p_n(\mathbf{x}) = \frac{k_n / n}{V_n}$$

- We hope that: $\lim_{n \rightarrow \infty} p_n(\mathbf{x}) = p(\mathbf{x})$
- To do this, we should have

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

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

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



n samples

Density Estimation

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

*What items can be controlled?
How?*

- Fix V_n and determine k_n
 - Parzen Windows

- Fix k_n and determine V_n
 - k_n -Nearest-Neighbor

Parzen Windows

$$p_n(\mathbf{x}) = \frac{k_n / n}{V_n} \quad \text{Fix } V_n \text{ and determine } k_n$$

- Assume R_n is a d-dimensional hypercube
- The length of each edge is h_n

$$V_n = h_n^d$$

Determine k_n with window function a.k.a. kernel function, potential function.

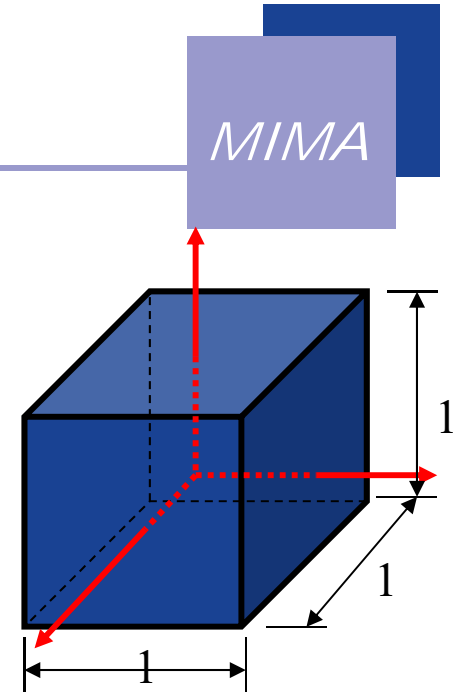


Emanuel Parzen
(1929-)

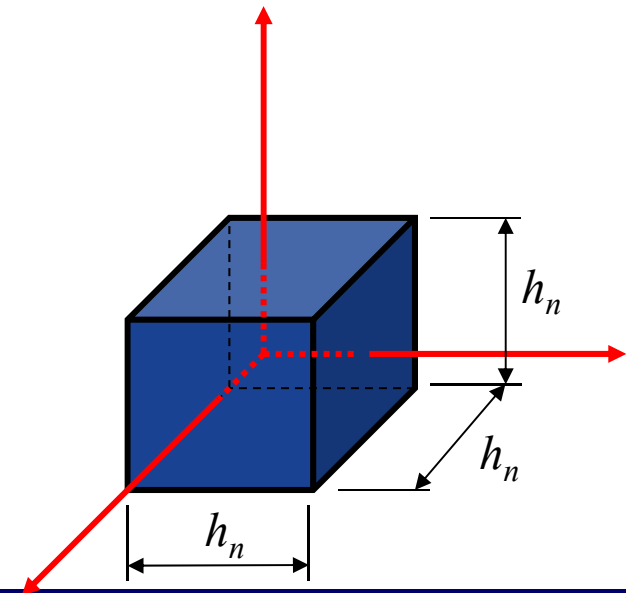
Window function

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

- It defines a unit hypercube centered at the origin.



$$\varphi\left(\frac{\mathbf{x}}{h_n}\right) = \begin{cases} 1 & |x| \leq h_n / 2 \\ 0 & \text{otherwise} \end{cases}$$

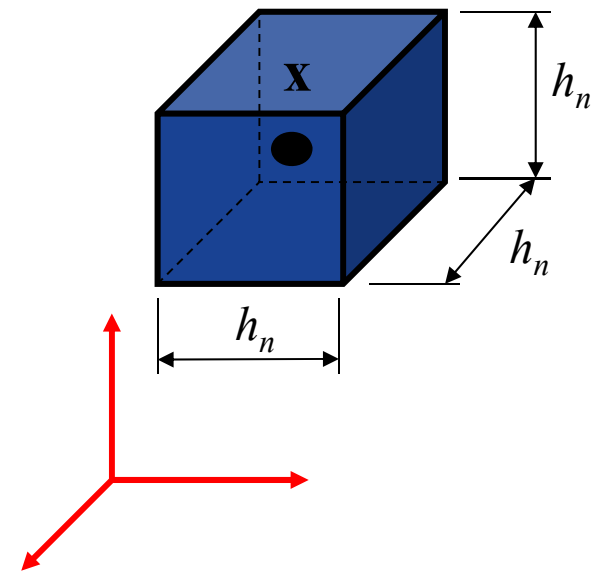


Window function

$$\varphi\left(\frac{\mathbf{x} - \mathbf{x}'}{h_n}\right) = \begin{cases} 1 & |x_j - x'_j| \leq h_n / 2 \\ 0 & \text{otherwise} \end{cases}$$


- 1 means that \mathbf{x}' falls within the hypercube of volume v_n centered at \mathbf{x} .
- k_n : # samples inside the hypercube centered at \mathbf{x} ,

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




Parzen Window Estimation

MIMA

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

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

Parzen pdf

- $\varphi(\mathbf{u})$ is not limited to be the hypercube window function defined previously. It could be any pdf function


$$\int \varphi(\mathbf{u}) d\mathbf{u} = 1$$

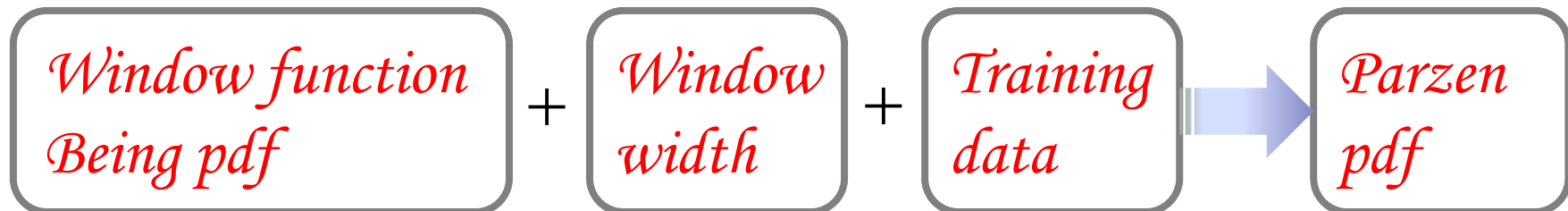
Parzen Window Estimation

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

- $p_n(\mathbf{x})$ is a pdf function?

Set $(\mathbf{x} - \mathbf{x}_i)/h_n = \mathbf{u}$.

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



Parzen Window Estimation

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

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

- $p_n(\mathbf{x})$: superposition (叠加) of n interpolations (插值)
- \mathbf{x}_i : contributes to $p_n(\mathbf{x})$ based on its “distance” from \mathbf{x} .

What is the effect of h_n (window width) on the Parzen pdf?



Parzen Window Estimation

- The effect of h_n

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

*Affects the amplitude
(vertical scale)*



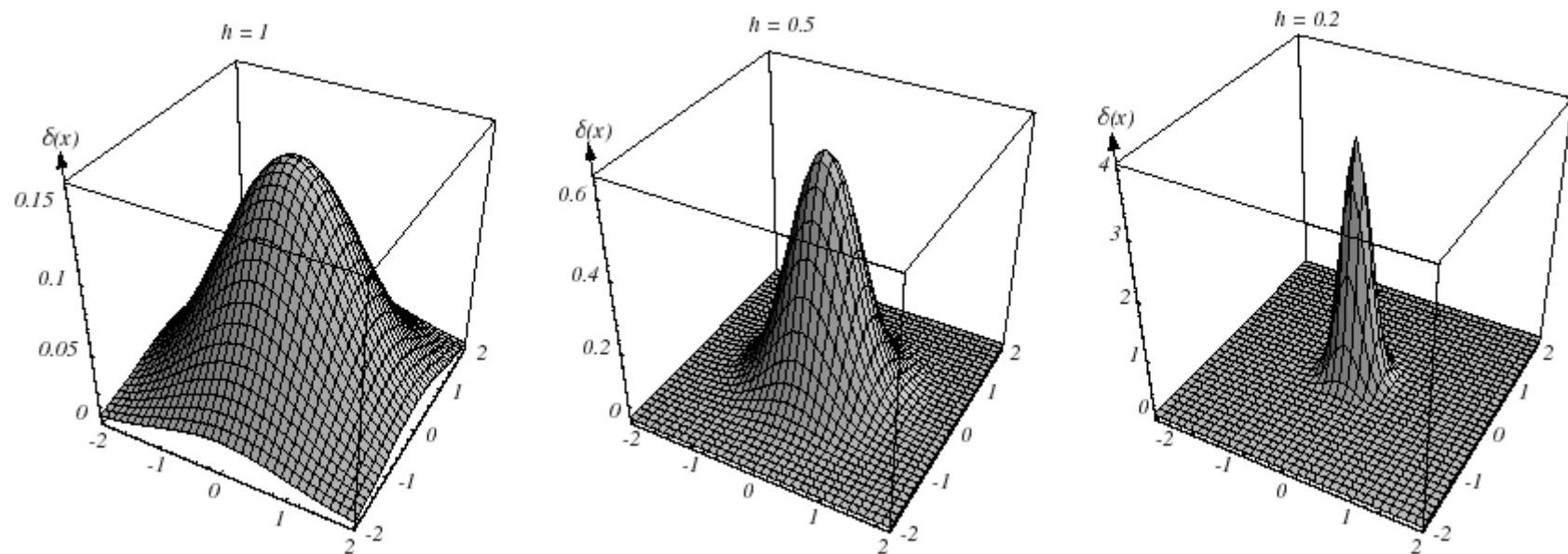
*Affects the width
(horizontal scale)*

Parzen Window Estimation

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

Suppose $\varphi(\cdot)$ being a 2-d Gaussian pdf.

The shape of $\delta_n(\mathbf{x})$ with decreasing values of h_n



Parzen Window Estimation

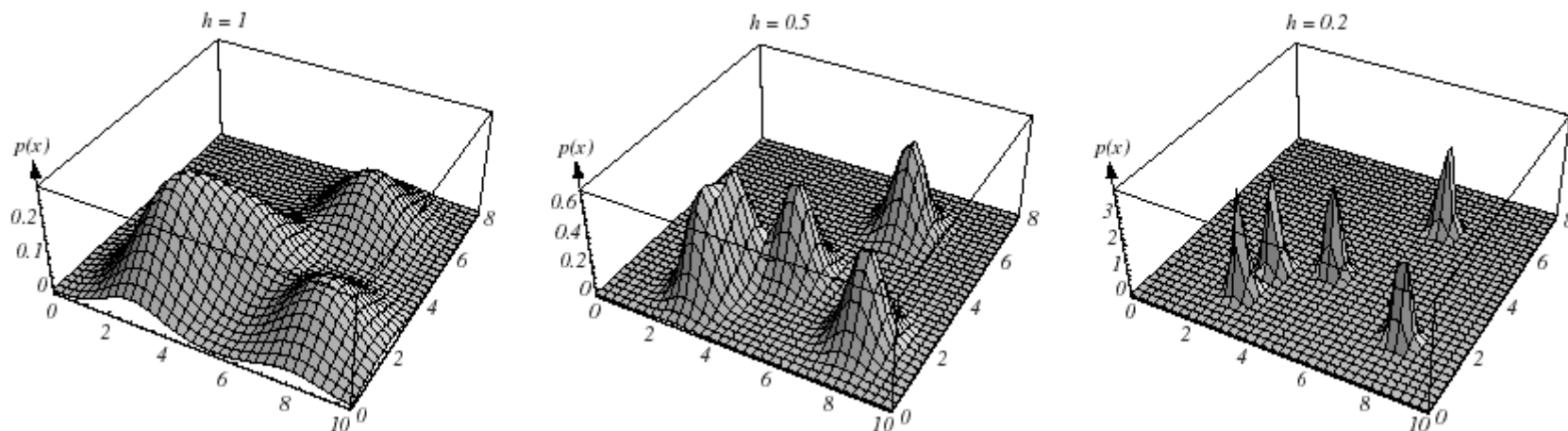
$$\delta_n(\mathbf{x}) = \frac{1}{V_n} \varphi\left(\frac{\mathbf{x}}{h_n}\right) \quad p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \delta_n(\mathbf{x} - \mathbf{x}_i)$$

- When h_n is very large, $\delta_n(x)$ will be broad with small amplitude.
 - $P_n(x)$ will be the superposition of n broad, slowly changing functions, i.e., being smooth with low resolution.
- When h_n is very small, $\delta_n(x)$ will be sharp with large amplitude.
 - $P_n(x)$ will be the superposition of n sharp pulses, i.e., being variable/unstable with high resolution.

Parzen Window Estimation

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

- *Parzen window estimations for five samples, supposing that $\varphi(\cdot)$ is a 2-d Gaussian pdf.*



Parzen Window Estimation

- Convergence conditions

- To ensure convergence, i.e.,

$$\lim_{n \rightarrow \infty} E[p_n(\mathbf{x})] = p(\mathbf{x}) \quad \lim_{n \rightarrow \infty} Var[p_n(\mathbf{x})] = 0$$

- We have the following additional **constraints**:

$$\sup_{\mathbf{u}} \varphi(\mathbf{u}) < \infty$$

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

$$\lim_{\|\mathbf{u}\| \rightarrow \infty} \varphi(\mathbf{u}) \prod_{i=1}^d u_i = 0$$

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

Illustrations

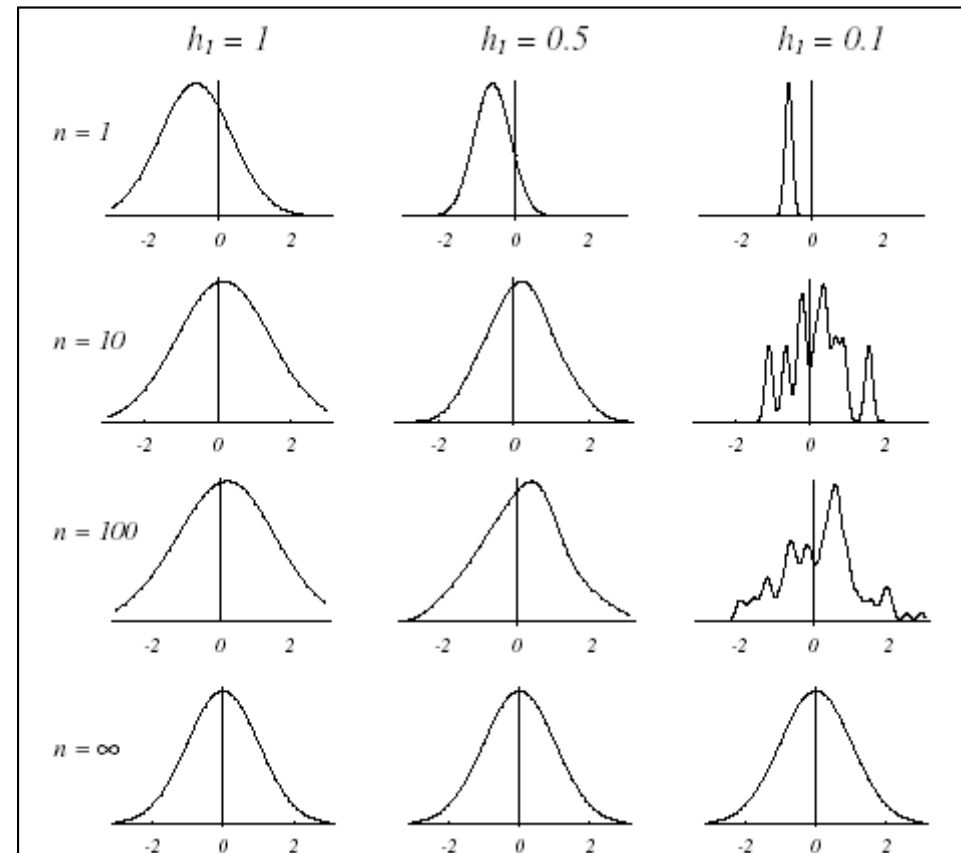
- One dimension case:

$$X \sim N(0,1)$$

$$\varphi(\mathbf{u}) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$$

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

$$h_n = h_1 / \sqrt{n}$$



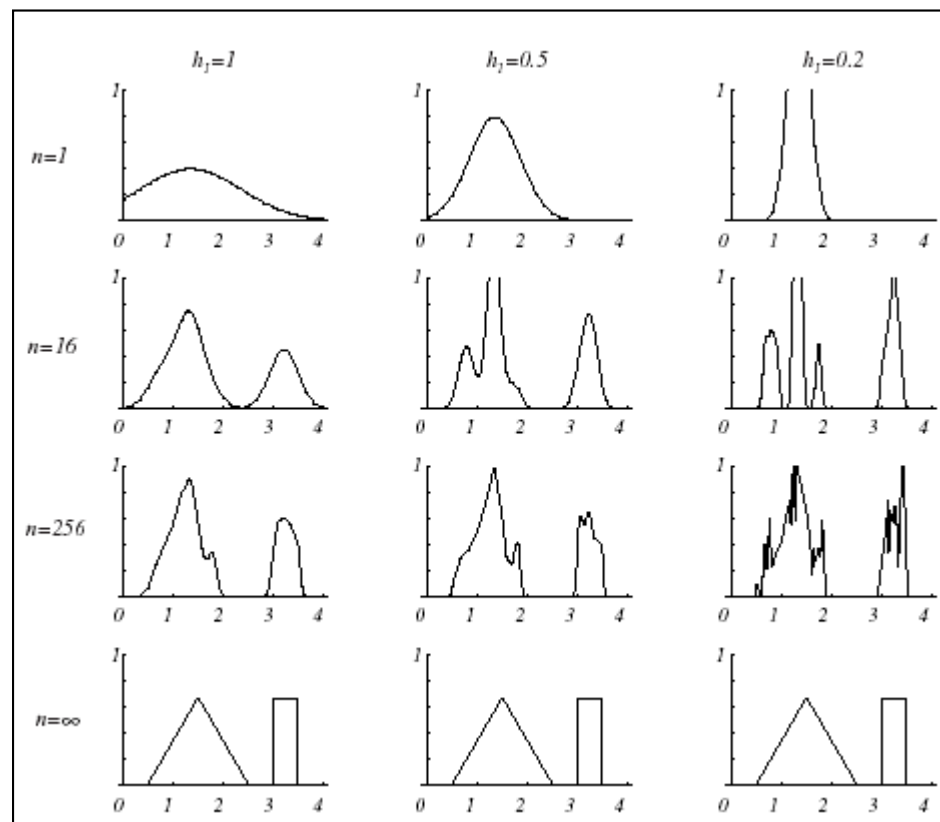
Illustrations

- One dimension case:

$$\varphi(\mathbf{u}) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$$

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

$$h_n = h_1 / \sqrt{n}$$

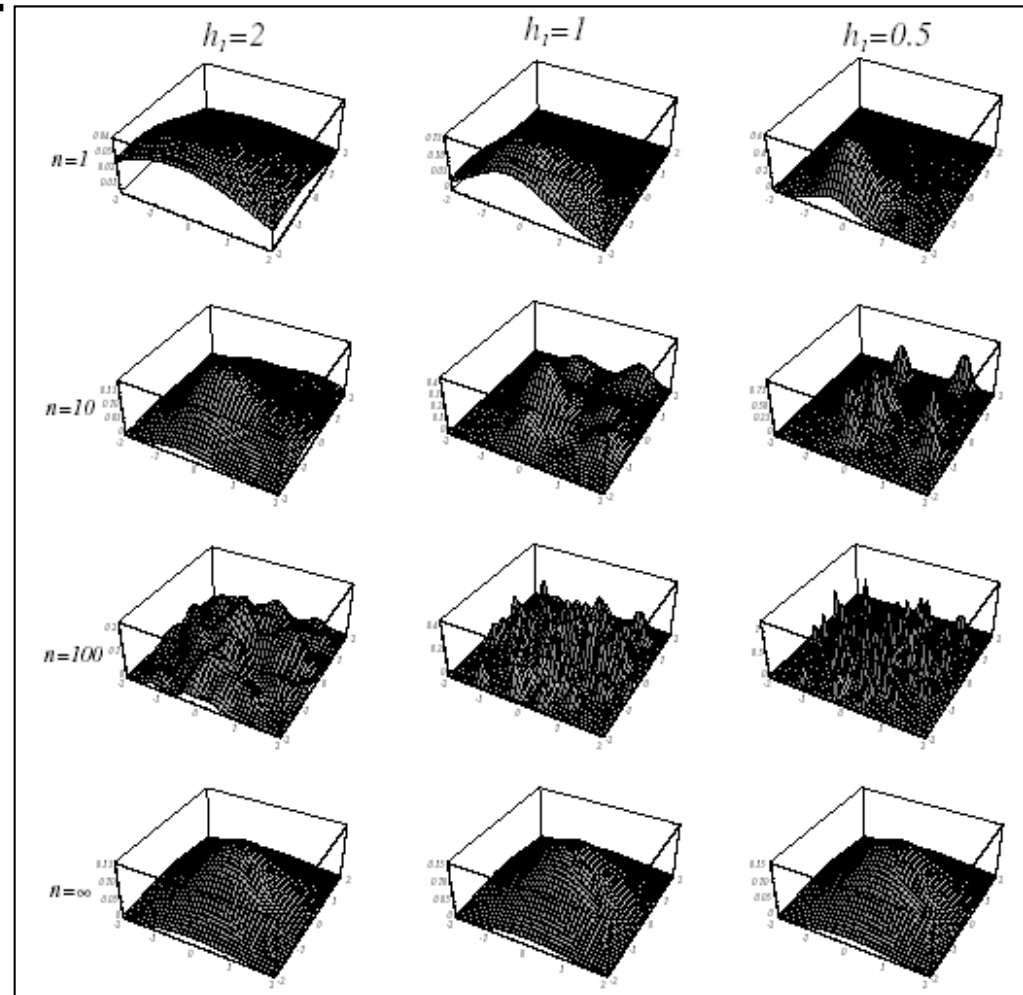


Illustrations

- Two dimension case:

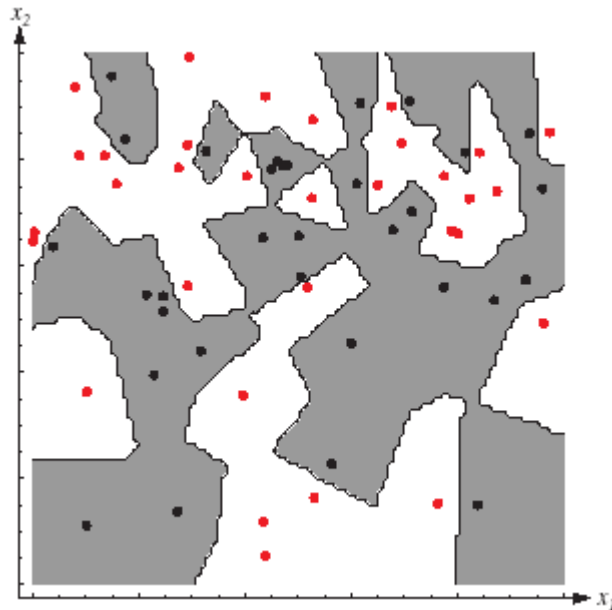
$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_n^2} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

$$h_n = h_1 / \sqrt{n}$$

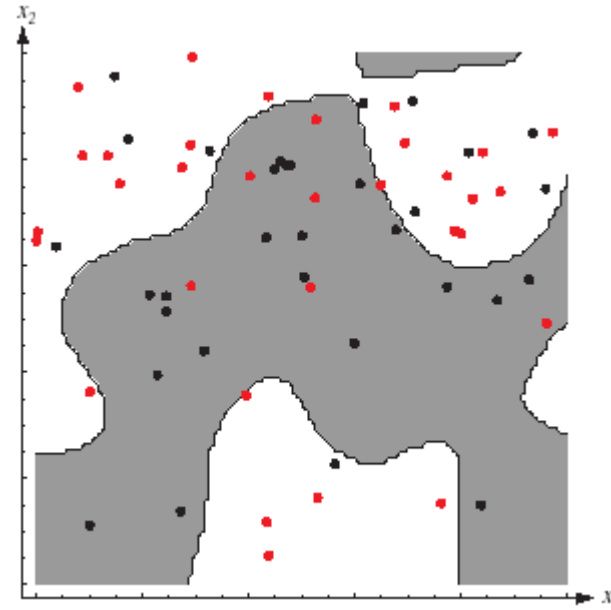


Classification Example

Smaller window



Larger window



Choosing Window Function

- V_n must approach zero when $n \rightarrow \infty$, but at a rate slower than $1/n$, e.g.,

$$V_n = V_1 / \sqrt{n}$$

- The value of initial volume V_1 is important.
- In some cases, a cell volume is proper for one region but unsuitable in a different region.

k_n -Nearest Neighbor

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

- Fix k_n and then determine V_n
- To estimate $p(\mathbf{x})$, we can center a cell about \mathbf{x} and let it grow until it captures k_n samples, k_n is some specified function of n , e.g.,

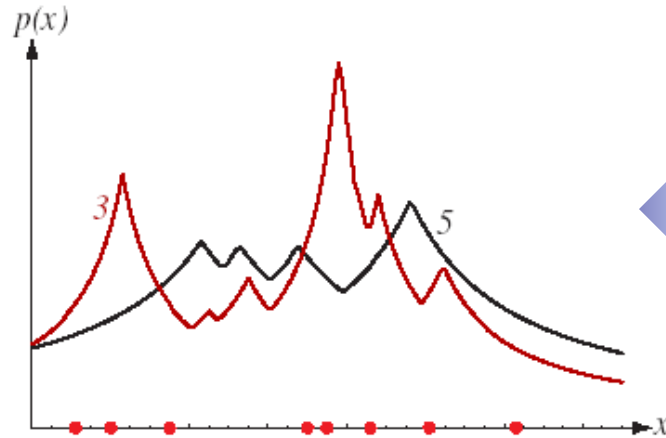
$$k_n = \sqrt{n}$$

Principled rule to choose k_n

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

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

k_n -Nearest Neighbor



Eight points in one dimension ($n=8, d=1$)

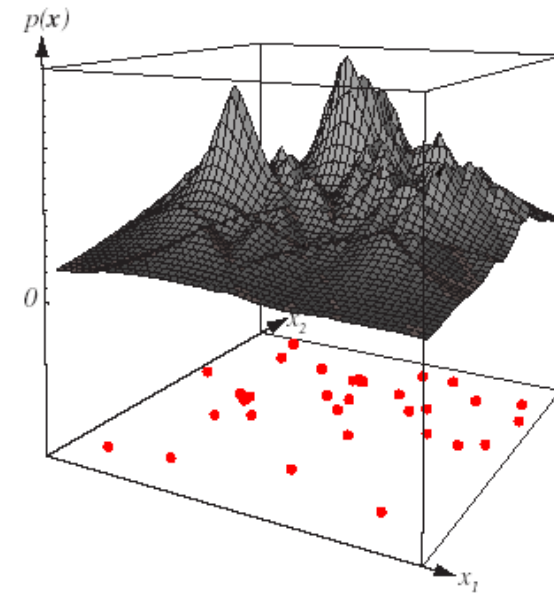
Red curve: $k_n=3$

Black curve: $k_n=5$

Thirty-one points in two dimensions

($n=31, d=2$)

Black surface: $k_n=5$



Estimation of A Posterior probability

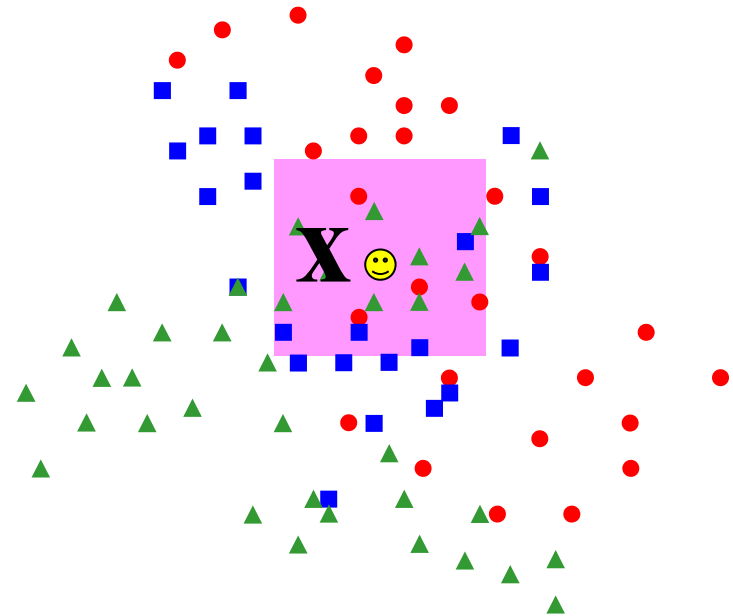
MIMA

$$P_n(\omega_i | \mathbf{X}) = ?$$

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

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

$$\sum_{j=1}^c p_n(x_n, \omega_j) = \frac{k_n / n}{V_n}$$



Estimation of A Posterior probability

$$P_n(\omega_i | \mathbf{x}) = ?$$

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

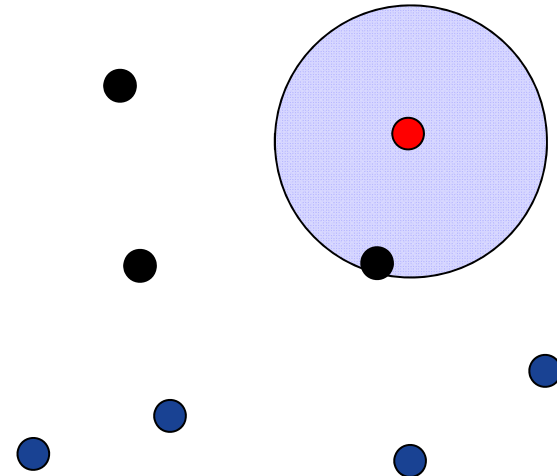
$$\left. \begin{aligned} p_n(x, \omega_i) &= \frac{k_i / n}{V_n} \\ \sum_{j=1}^c p_n(x_n, \omega_j) &= \frac{k_n / n}{V_n} \end{aligned} \right\} \begin{aligned} &\text{The value of } V_n \text{ or } k_n \text{ can be} \\ &\text{determined base on Parzen window} \\ &\text{or } k_n\text{-nearest-neighbor technique.} \end{aligned}$$

Nearest Neighbor Classifier

MIMA

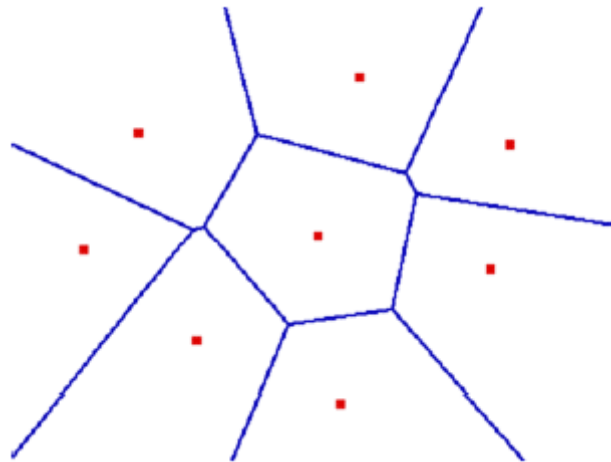
- Store all training examples
- Given a new example x to be classified, search for the training example (x_i, y_i) whose x_i is most similar (or closest) to x , and predict y_i . (Lazy Learning)

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



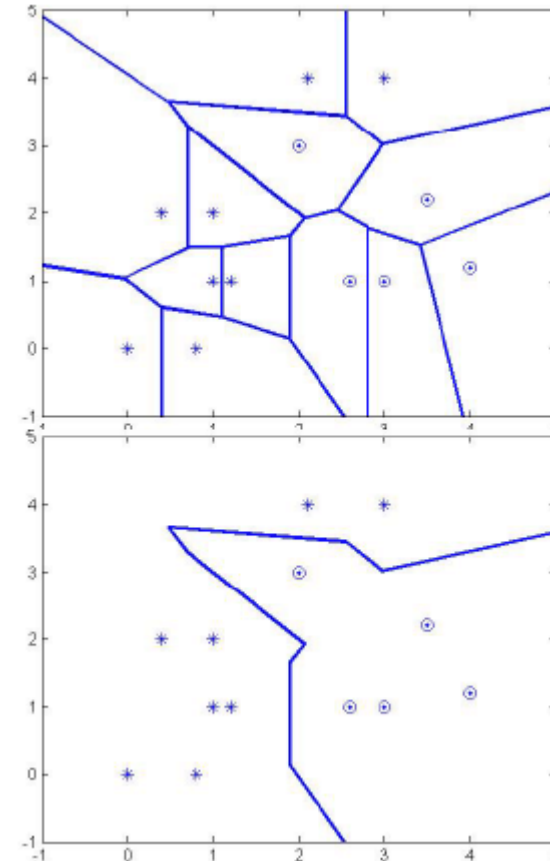
Decision Boundaries

- Decision Boundaries
 - The voronoi diagram
 - Given a set of points, a Voronoi diagram describes the areas that are nearest to any given point.
 - These areas can be viewed as zones of control.

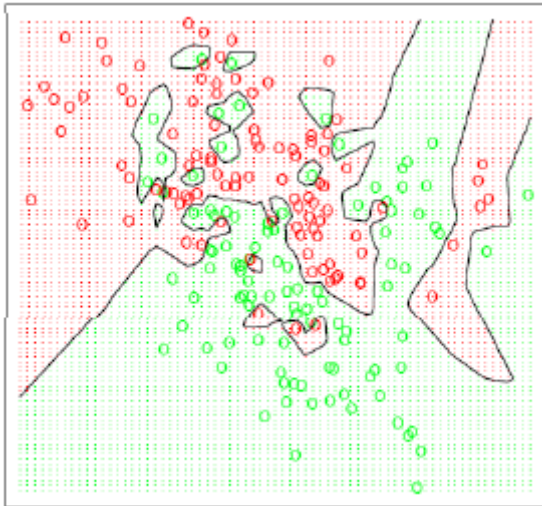


Decision Boundaries

- Decision boundary is formed by only retaining these line segment separating different classes.
- The more training examples we have stored, the more complex the decision boundaries can become.



Decision Boundaries

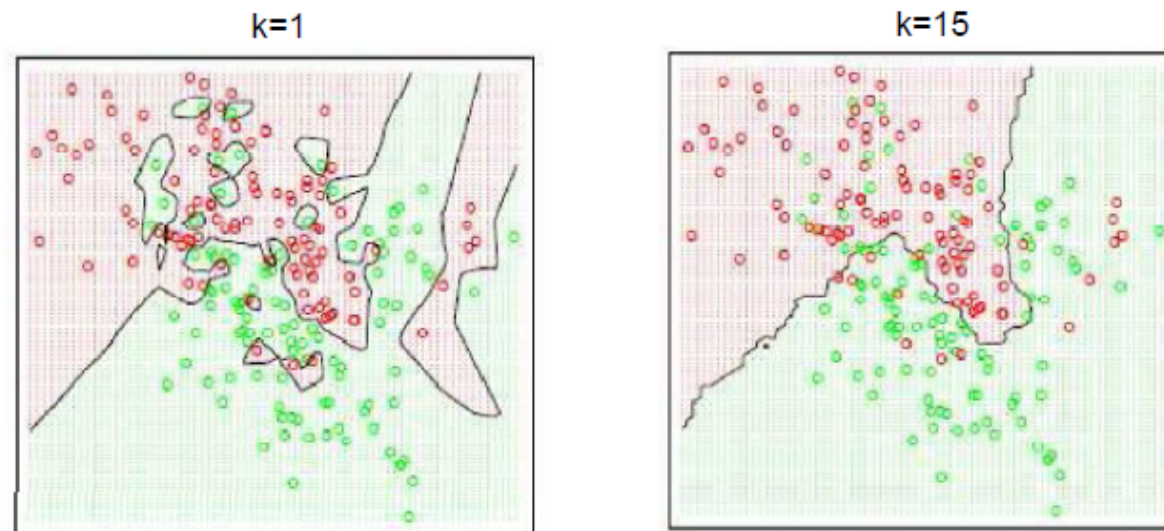


- With large number of examples and noise in the labels, the decision boundary can become nasty!
- It can be bad some times-note the islands in this figure, they are formed because of noisy examples.
- If the nearest neighbor happens to be a noisy point, the prediction will be incorrect.

How to deal with this?

Effect of k

- Different k values give different results:
 - Large k produces smoother boundaries
 - The impact of class label noises canceled out by one another.
 - When k is too large, what will happen.
 - Oversimplified boundaries, e.g., $k=N$, we always predict the majority class



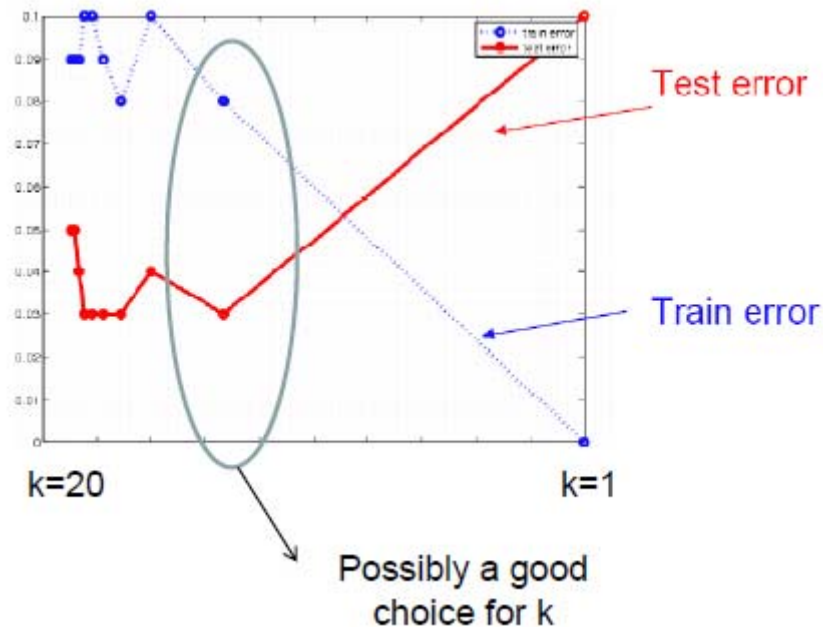
Figures from Hastie, Tibshirani and Friedman (Elements of Statistical Learning)

How to Choose k ?

- Can we choose k to minimize the mistakes that we make on training examples? (training error)
 - What is the training error of nearest-neighbor?
- Can we choose k to minimize the mistakes that we make on test examples? (test error)

How to Choose k ?

- How do training error and test error change as we change the value of k ?



Model Selection

- Choosing k for k -NN is just one of the many model selection problems we face in machine learning.
- Model selection is about choosing among different models
 - Linear regression vs. quadratic regression
 - K -NN vs. decision tree
 - Heavily studied in machine learning, crucial importance in practice.
- If we use training error to select models, we will always choose more complex ones.

Model Selection

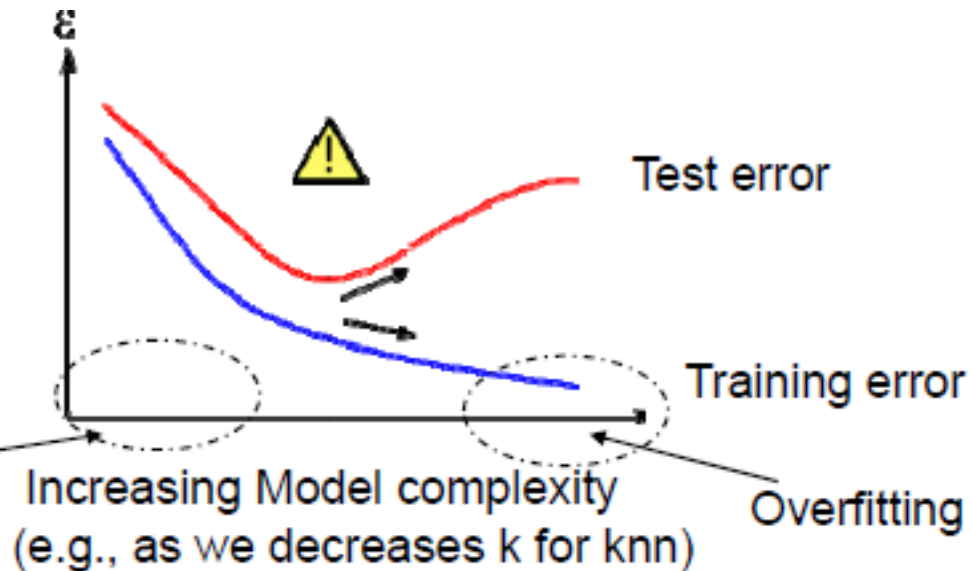
- Choosing k for k -NN is just one of the many model selection problems we face in machine learning.

- Model complexity difference

-
-

- underfitting

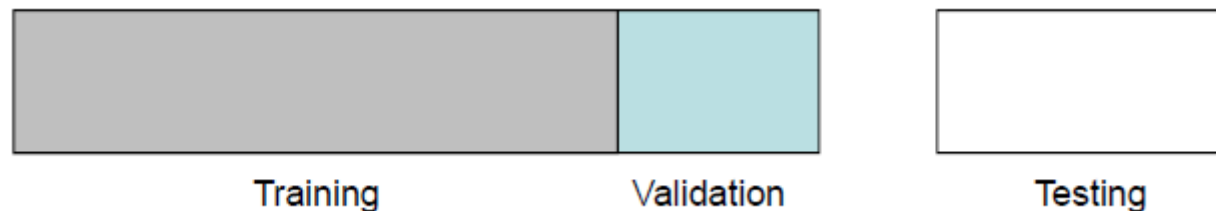
Increasing Model complexity
(e.g., as we decrease k for k nn)
importance in practice.



- If we use training error to select models, we will always choose more complex ones.

Model Selection

- We can keep part of the labeled data apart as validation data.
- Evaluate different k values based on the prediction accuracy on the validation data
- Choose k that minimize validation error



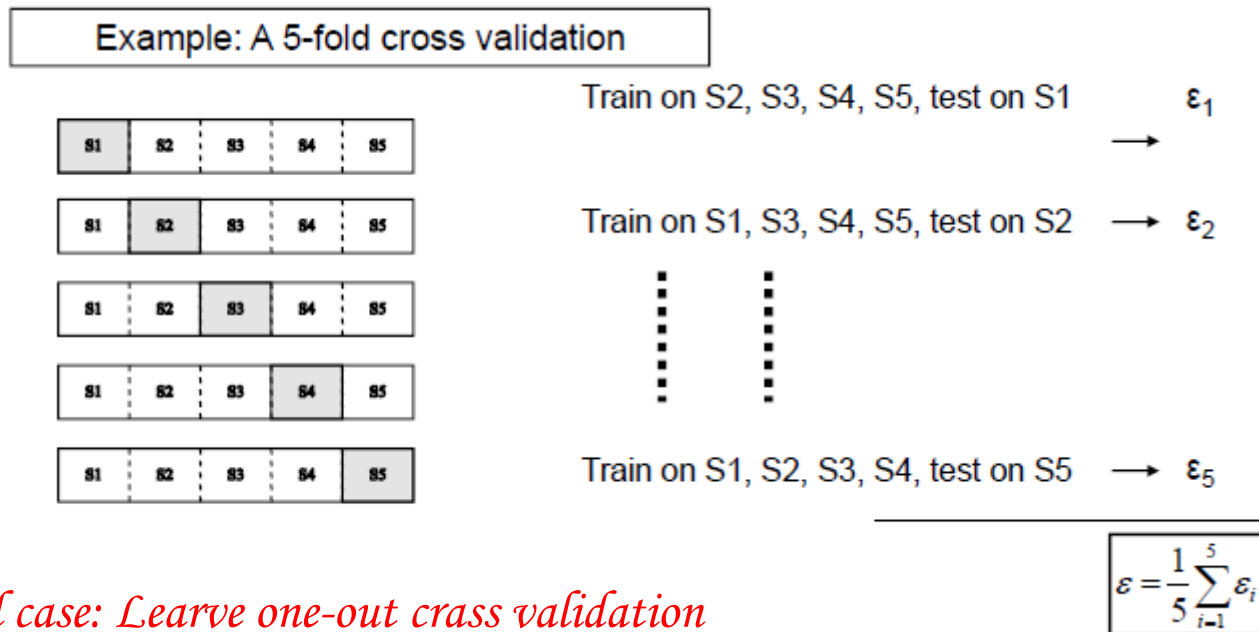
Validation can be viewed as another name for testing, but the name testing is typically reserved for final evaluation purpose, whereas validation is mostly used for model selection purpose.

Model Selection

- The impact of validation set size
 - If we only reserve one point in our validation set, should we trust the validation error as a reliable estimate of our classifier's performance?
 - The larger the validation set, the more reliable our model selection choices are
 - When the total labeled set is small, we might not be able to get a big enough validation set – leading to unreliable model selection decisions

Model Selection

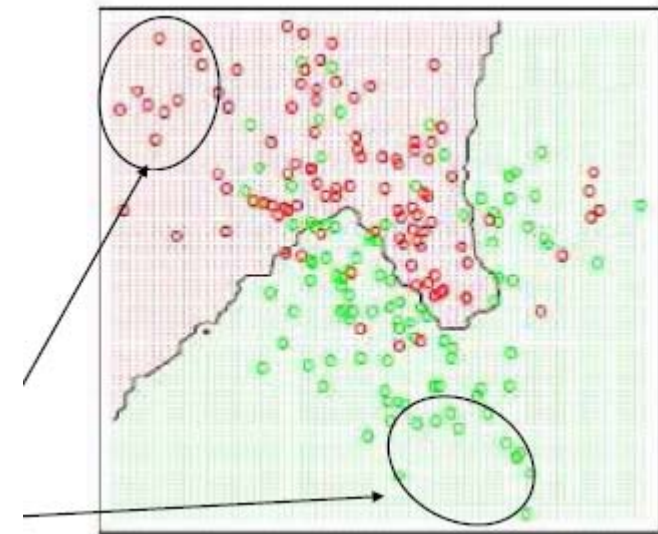
- K-fold Cross Validation
 - Perform learning/testing K times
 - Each time reserve one subset for validation set, train on the rest



Special case: Leave one-out crass validation

Other issues of kNN

- It can be computationally expensive to find the nearest neighbors!
 - Speed up the computation by using smart data structures to quickly search for approximate solutions
- For large data set, it requires a lot of memory
 - Remove unimportant examples



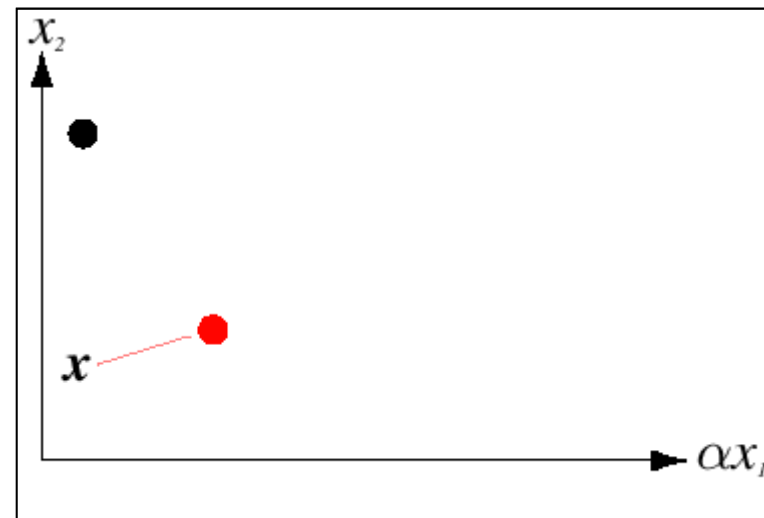
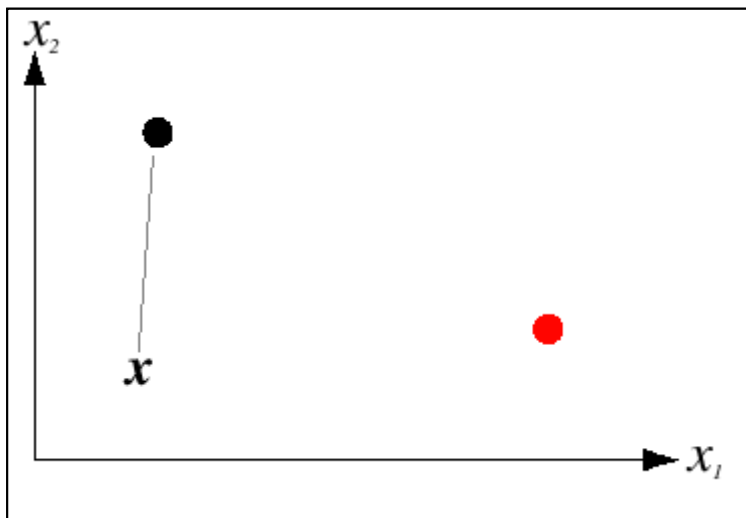
Final words on KNN

- KNN is what we call *lazy learning* (vs. *eager learning*)
 - Lazy: learning only occur when you see the test example
 - Eager: learn a model before you see the test example, training examples can be thrown away after learning
- Advantage:
 - Conceptually simple, easy to understand and explain
 - Very flexible decision boundaries
 - Not much learning at all!
- Disadvantage
 - – It can be hard to find a good distance measure
 - – Irrelevant features and noise can be very detrimental
 - – Typically can not handle more than 30 attributes
 - – Computational cost: requires a lot computation and memory

Distance Metrics

- Distance Measurement is an importance factor for nearest-neighbor classifier, e.g.,
 - To achieve *invariant pattern recognition and data mining results*.

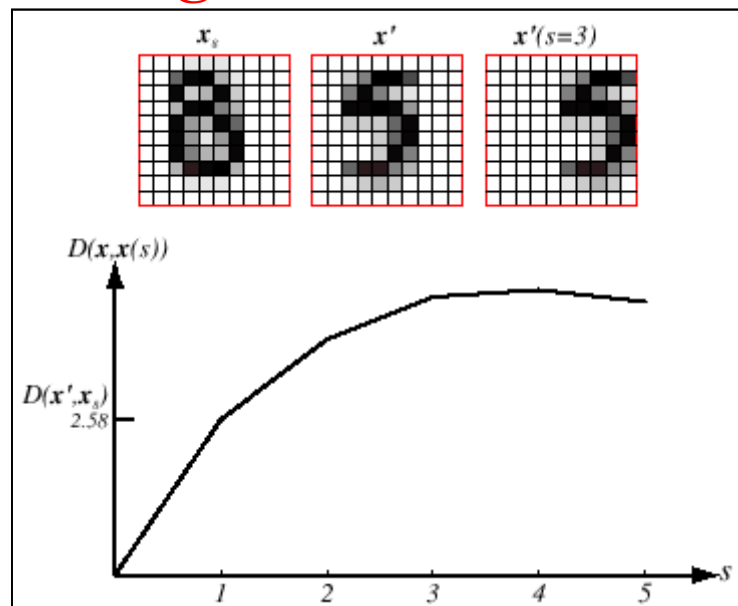
The effect of change units



Distance Metrics

- Distance Measurement is an importance factor for nearest-neighbor classifier, e.g.,
 - To achieve *invariant pattern recognition and data mining results*.

The effect of change units



Properties of a Distance Metric

- Nonnegativity

$$D(\mathbf{a}, \mathbf{b}) \geq 0$$

- Reflexivity

$$D(\mathbf{a}, \mathbf{b}) = 0 \text{ iff } \mathbf{a} = \mathbf{b}$$

- Symmetry

$$D(\mathbf{a}, \mathbf{b}) = D(\mathbf{b}, \mathbf{a})$$

- Triangle Inequality

$$D(\mathbf{a}, \mathbf{b}) + D(\mathbf{b}, \mathbf{c}) \geq D(\mathbf{a}, \mathbf{c})$$

Minkowski Metric (L_p Norm)

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^d |x_i|^p \right)^{1/p}$$

■ 1. L_1 norm

Manhattan or city block distance

$$L_1(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_1 = \sum_{i=1}^d |a_i - b_i|$$

■ 2. L_2 norm

Euclidean distance

$$L_2(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_2 = \left(\sum_{i=1}^d |a_i - b_i|^2 \right)^{1/2}$$

■ 3. L_∞ norm

Chessboard distance

$$L_\infty(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_\infty = \left(\sum_{i=1}^d |a_i - b_i|^\infty \right)^{1/\infty} = \max(|a_i - b_i|)$$

Minkowski Metric (L_p Norm)

$$\left(\sum_{i=1}^u |a_i - b_i|^p \right)^{1/p}$$

- 1. L_1 norm ∞

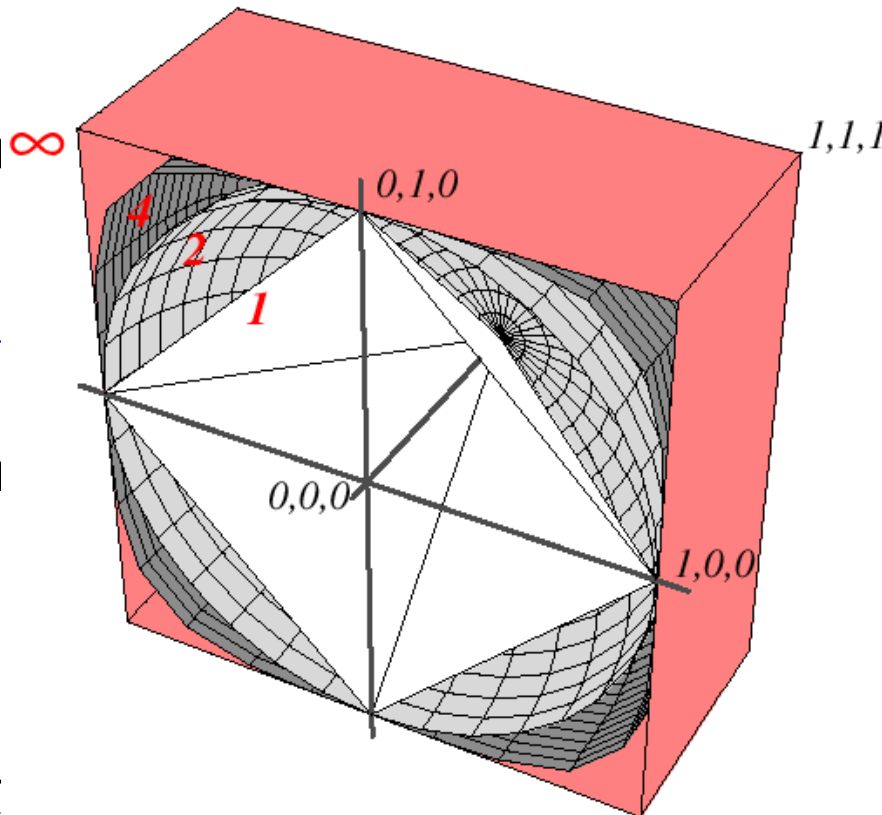
*Manhattan
block distance*

- 2. L_2 norm

*Euclidean
distance*

- 3. L_∞ norm

*Chessboard
distance*



$$|a_i - b_i|$$

$$\left(\sum_{i=1}^u |a_i - b_i|^2 \right)^{1/2}$$

$$L_\infty(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_\infty = \left(\sum_{i=1}^u |a_i - b_i|^\infty \right)^{1/\infty} = \max(|a_i - b_i|)$$

Summary

- Basic setting for non-parametric techniques
 - Let the data speak for themselves
 - Parametric form not assumed for class-conditional pdf
 - Estimate class-conditional pdf from training examples
 - Make predictions based on Bayes Theorem
- Fundamental results in density estimation

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

Summary

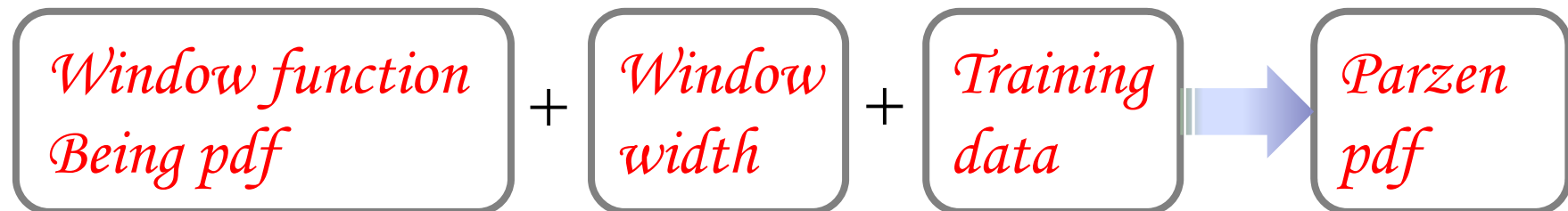
$$p_n(\mathbf{x}) = \frac{k_n / n}{V_n}$$

MIMA

■ Parzen Windows

- Fix V_n and then determine k_n

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



Summary

- k_n -Nearest-Neighbor
 - Fix k_n and then determine V_n
- Fix k_n and then determine V_n
- To estimate $p(\mathbf{x})$, we can center a cell about \mathbf{x} and let it grow until it captures k_n samples, where is some specified function of n , e.g.,

$$k_n = \sqrt{n}$$

Principled rule to choose k_n

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

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

MIMA Group

[Thank You !]

Any Question?