パターン認識
Pattern Recognition

佐藤真一
Shin'ichi Satoh

国立情報学研究所
National Institute of Informatics

May 2, 2023

Course Information

- Course web page: https://research.nii.ac.jp/~satoh/utpr/
- Course materials can be found in the above page
- Course videos can be found in ITC-LMS
- Credits will be given based on final report (mandatory)
- Attendance record will not be taken
- Assignments may be imposed (3 out of 7 are mandatory: subject to change)
- The first assignment issued on April 18 will be due on today, the second issued on April 25 will be due on next week, and the third issued today will be due on the next to next week (May 16)
- If you fail to submit minimum 3 assignments and final report, you will not obtain credits.

Recap

- We visited "parametric" methods so far.
- Probability distribution functions (or equivalently decision boundaries) can be represented by parametric forms.
- e.g., Normal density case: mean and variance (or covariance matrix)
- These methods assume that the underlying probability distribution of the actual observations is known and yields parametric forms.
- However, in many cases this assumption is suspect.

Today's topics

- Nonparametric Density Estimation Methods
  - Parzen Window
  - k-Nearest Neighbor Estimation

- Simple approach is to compose histogram
- Knowing sample data, we can compose histogram with certain bin size (division of each axis)
- Treat the histogram as probability distribution function

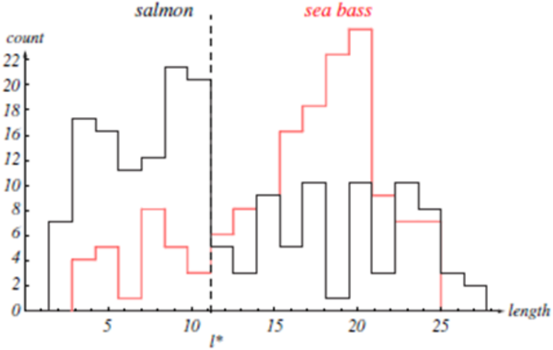
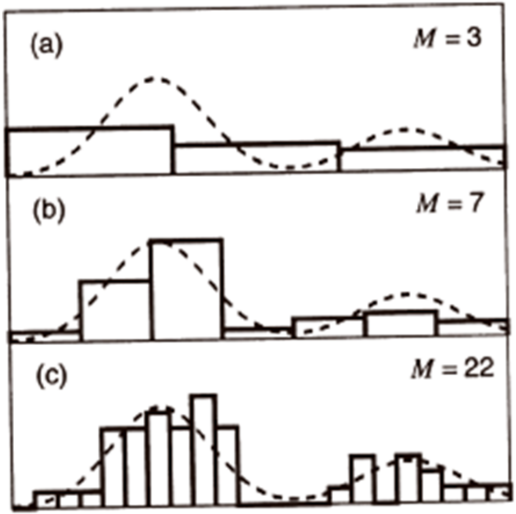


FIGURE 1.2. Histograms for the length feature for the two categories. No single threshold value of the length will serve to unambiguously discriminate between the two categories; using length alone, we will have some errors. The value marked l^* will lead to

Nonparametric Methods

- The optimal number of bins M (or bin size) is the issue.
  - If bin width is small (i.e., big M), then the estimated density is very spiky (i.e., noisy).
  - If bin width is large (i.e., small M), then the true structure of the density is smoothed out.
- In practice, we need to find an optimal value for M that compromises between these two issues.
- Also, how we extend to the multidimensional case?



Nonparametric Density Estimation

- The probability that a given vector $\mathbf{x}$, drawn from the unknown density $p(\mathbf{x})$, will fall inside some region R in the input space is given by:

$$P = \int_R p(\mathbf{x}')d\mathbf{x}'$$

- If we have n data points $\{x_1, x_2, \ldots, x_n\}$ drawn independently from $p(\mathbf{x})$, the probability that k of them will fall in R is given by the binomial law:

$$P(k) = P_k = \binom{n}{k} P^k(1-P)^{n-k}$$

Nonparametric Density Estimation

- The expected value of k is:

$$E\{k\} = nP$$

- The expected percentage of points falling in R is:

$$E\{\frac{k}{n}\} = P$$

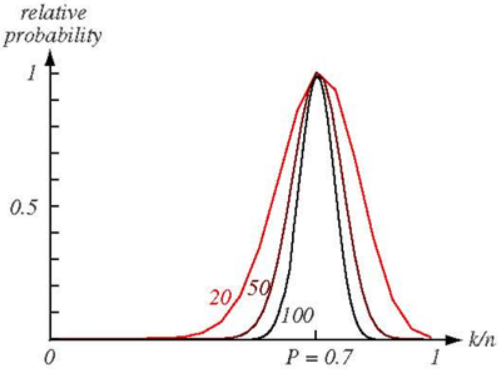
- The variance is given by:

$$Var\{\frac{k}{n}\} = E\{(\frac{k}{n} - P)^2\} = \frac{P(1-P)}{n}$$

The distribution is sharply peaked as $n \to \inf$, thus:

$$P \approx \frac{k}{n}$$

$\to$ Approximation 1

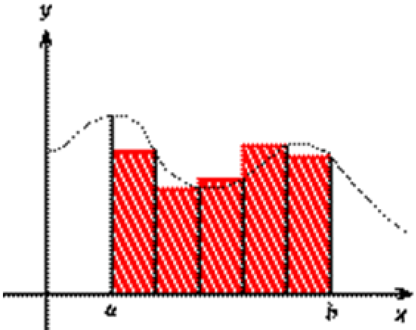


Nonparametric Density Estimation

If we assume that $p(\mathbf{x})$ is continuous and does not vary significantly over the region R, we can approximate P by:

$$P = \int_R p(\mathbf{x}')d\mathbf{x}' \approx p(\mathbf{x})V$$

$\rightarrow$ Approximation 2
where V is the volume enclosed by R.



Nonparametric Density Estimation

- Combining these two approximations we have:

$$\boxed{p(\mathbf{x}) \approx \frac{k/n}{V}}$$

- The above approximation is based on contradictory assumptions:
  - R is relatively **large** (i.e., it contains many samples so that P_k is sharply peaked): Approximation 1
  - R is relatively **small** so that $p(x)$ is approximately constant inside the integration region: Approximation 2
- We need to choose an optimum R in practice ...

Nonparametric Density Estimation

- Suppose we form regions $R_1, R_2, \ldots$ containing $\mathbf{x}$.
  - R_1 contains k_1 sample, R_2 contains k_2 samples, etc.
  - We assume that each case corresponds to n samples.
- R_i has volume V_i and contains k_i samples.
- The n-th estimate $p_n(\mathbf{x})$ of $p(\mathbf{x})$ is given by:

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

Nonparametric Density Estimation

The following conditions must be satisfied in order for $p_n(\mathbf{x})$ to converge to $p(\mathbf{x})$:

$$\lim_{n\to\infty} V_n = 0 \quad \text{Approximation 1}$$

$$\lim_{n\to\infty} k_n = \infty \quad \text{Approximation 2}$$

$$\lim_{n\to\infty} \frac{k_n}{n} = 0 \quad \text{to allow } p_n(\mathbf{x}) \text{ to converge}$$

Nonparametric Density Estimation

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

How to choose the optimum values for V_n and k_n?
Two leading approaches:

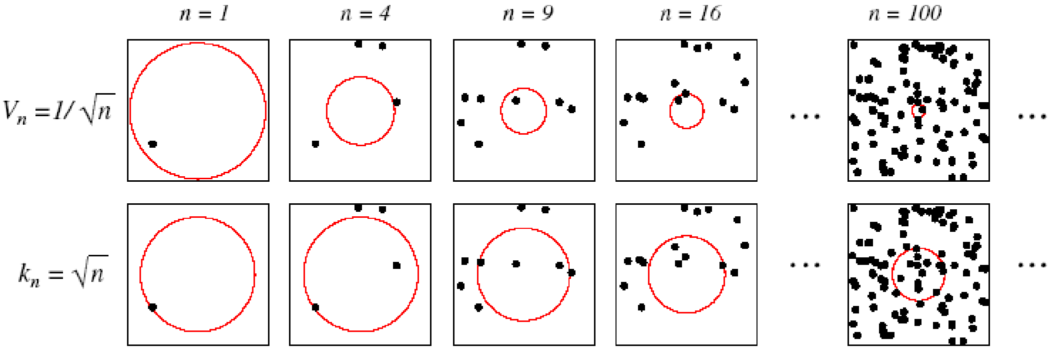
(1) Fix the volume V_n and determine k_n from the data (kernel-based density estimation methods), e.g.,

$$V_n = \frac{1}{\sqrt{n}}$$

(2) Fix the value of k_n and determine the corresponding volume V_n from the data (k-nearest neighbor method), e.g.,

$$k_n = \sqrt{n}$$

Nonparametric Density Estimation



Parzen Windows

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

- **Problem**: Given a vector $\mathbf{x}$, estimate $p(\mathbf{x})$
- Assume R_n to be a hypercube with sides of length h_n, centered on the point $\mathbf{x}$:

$$V_n = {h_n}^d$$

- To find an expression for k_n (i.e., # points in the hypercube) let us define a kernel function:

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

Parzen Windows

- The total number of points x_i falling inside the hypercube is:

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

- Then, the estimate

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

becomes

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

$\rightarrow$ Parzen windows estimate

Parzen Windows

- The density estimate is a superposition of kernel functions and the samples x_i.

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

- $\varphi(u)$ interpolates the density between samples.
- Each sample x_i contributes to the estimate based on its distance from $\mathbf{x}$.

Parzen Windows

- The kernel function $\varphi(u)$ can have a more general form (i.e., not just hypercube).
- In order for $p_n(\mathbf{x})$ to be a legitimate estimate, $\varphi(u)$ must be a valid density itself:

$$\varphi(u) \geq 0$$

$$\int \varphi(u) du = 1$$

Parzen Windows

The parameter h_n acts as a smoothing parameter that needs to be optimized.

- When h_n is too large, the estimated density is over-smoothed (i.e., superposition of "broad" kernel functions).
- When h_n is too small, the estimate represents the properties of the data rather than the true density (i.e., superposition of "narrow" kernel functions)

$\varphi(u)$ assuming different h_n values:

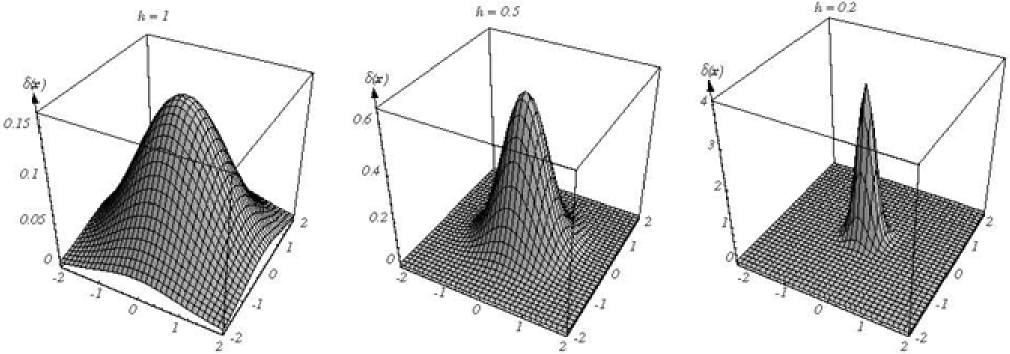


FIGURE 4.3. Examples of two-dimensional circularly symmetric normal Parzen windows for three different values of h. Note that because the $\delta(\mathbf{x})$ are normalized, different vertical scales must be used to show their structure. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example: $p_n(\mathbf{x})$ estimates assuming 5 samples:

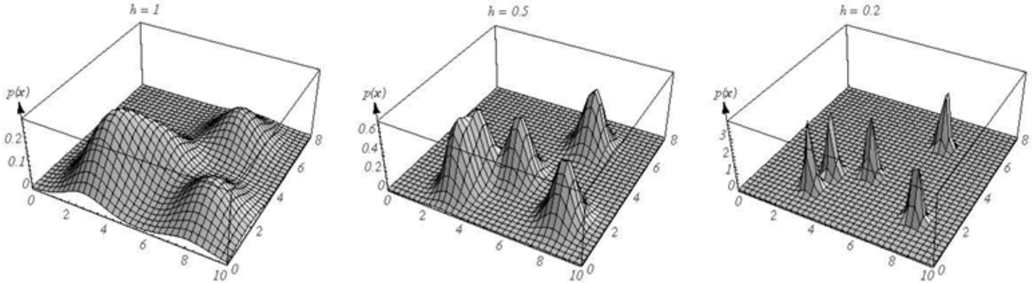
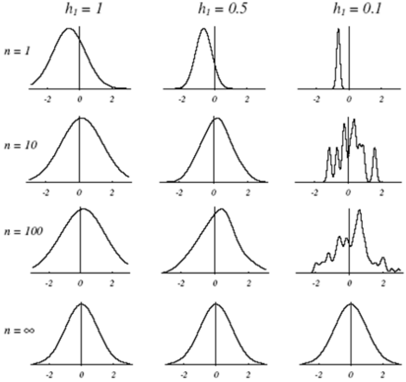


FIGURE 4.4. Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.3. As before, the vertical axes have been scaled to show the structure of each distribution. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

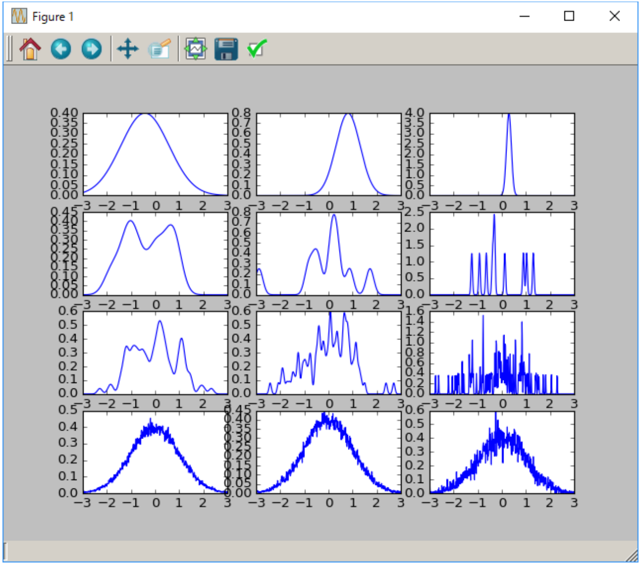
Parzen Windows

Example: both $p(\mathbf{x})$ and $\varphi(u)$ are Gaussian

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



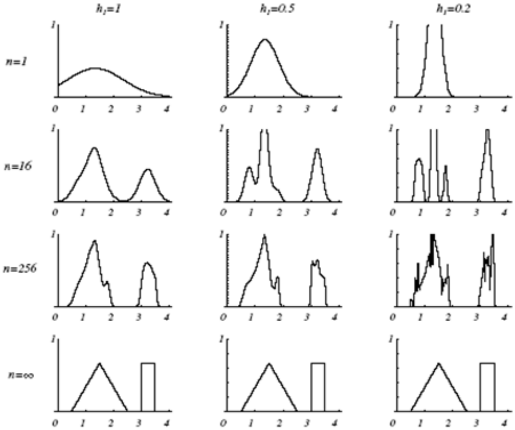
Exercise (parzeng.py)



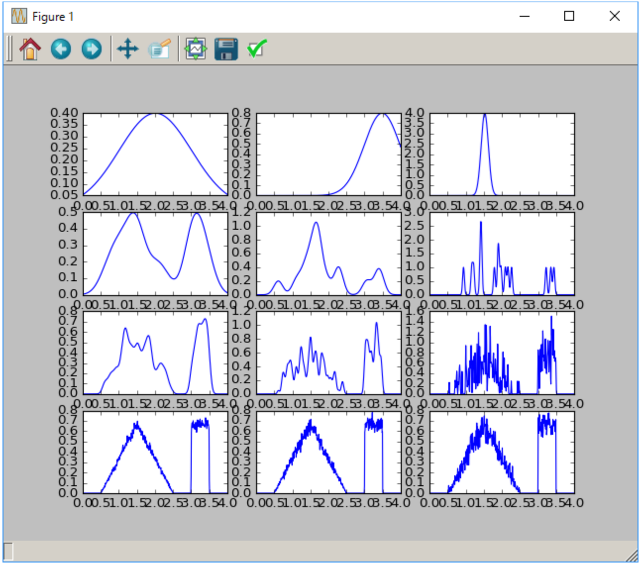
Parzen Windows

Example: $p(\mathbf{x})$ consists of a uniform and triangular density and $\varphi(u)$ is Gaussian

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



Exercise (parzentr.py)



k-Nearest Neighbor Estimate

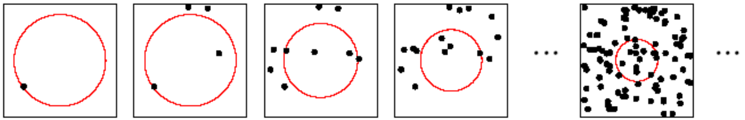
Fix k_n and allow V_n to vary:

- Consider a hypersphere around $\mathbf{x}$.
- Allow the radius of the hypersphere to grow until it contains k_n data points.
- V_n is determined by the volume of the hypersphere.

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

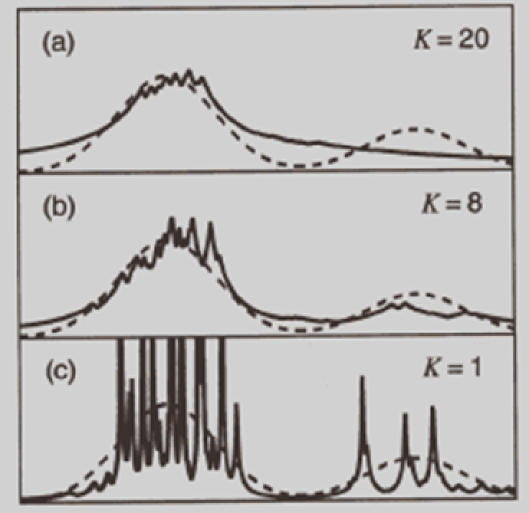
The size depends on the density

$k_n = \sqrt{n}$



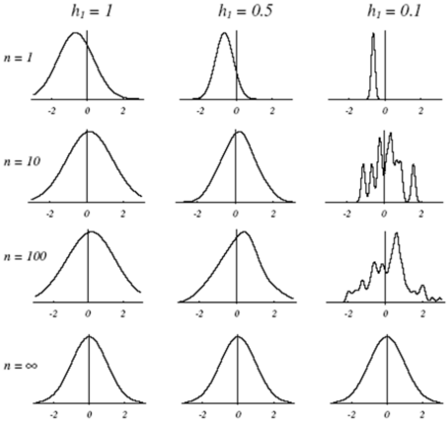
k-Nearest Neighbor Estimate

The parameter k_n acts as a smoothing parameter and needs to be optimized.

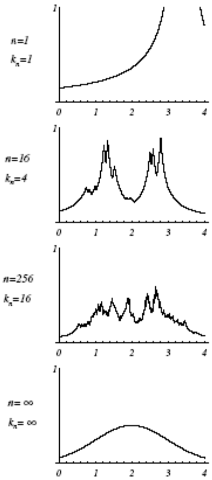


k-Nearest Neighbor Estimate

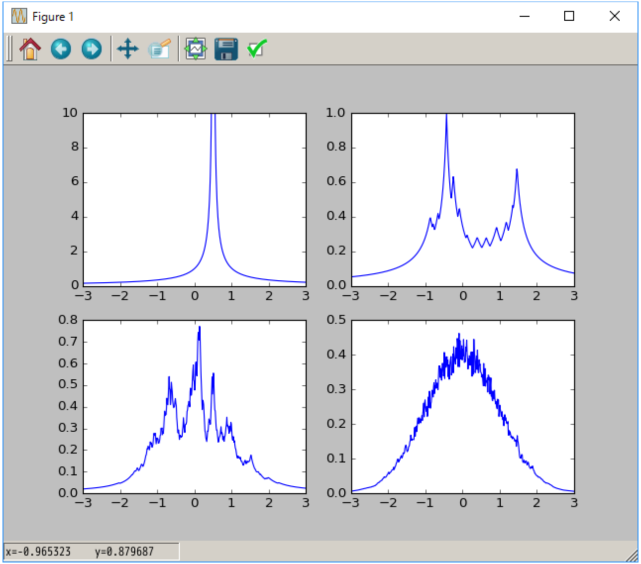
Parzen windows



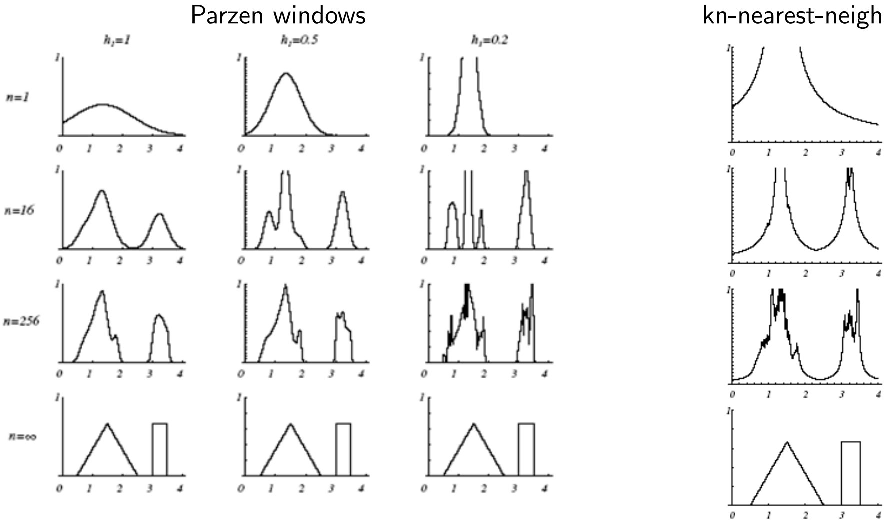
kn-nearest-neighbor



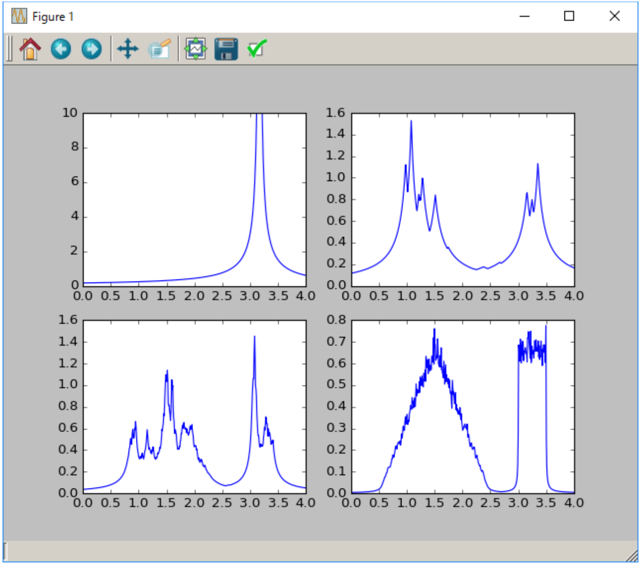
Exercise (knng.py)



k-Nearest Neighbor Estimate



Exercise (knntr.py)



Assignment

- Programming project and non-programming project are imposed.
- You are expected to solve either programming project OR non-programming project.
- Programming project is recommended.
- Of course you are most welcomed to solve both.
- Due on May 16.

Programming project

- Download data file from the course web site. The file contains two variables: x1 and x2.
- (The data is in Matlab format. Use "loadmatfile" for Scilab or "scipy.io.loadmat" for python. Refer to the course material on April 13)
- Assume that they are samples of two classes c1 and c2.
- Plot conditional probability distributions $p(\mathbf{x}|c_i)$ using Parzen windows (both Gaussian and box functions) and k-NN (with various k).
- Plot posterior probabilities $P(c_i|x)$ assuming prior probabilities $P(c_1) = P(c_2) = \frac{1}{2}$.

Non-programming project

- The probability that a given vector $\mathbf{x}$, drawn from the unknown density $p(\mathbf{x})$, will fall inside some region R in the input space is assumed to be P.
- If we have n data points $\{x_1, x_2, \ldots, x_n\}$ drawn independently from $p(\mathbf{x})$, we assume that k of the points will fall in R.
- Show that the expected value of k is:

$$E\{k\} = nP$$

- Show that the expected percentage of points falling in R is:

$$E\{\frac{k}{n}\} = P$$

- Show that the variance is given by:

$$Var\{\frac{k}{n}\} = E\{(\frac{k}{n} - P)^2\} = \frac{P(1-P)}{n}$$