Pattern Recognition[1]

Dokuz Eylül University
Computer Engineering Department
19 February 2001 – 8 June 2001

Akira Imada
imada@cs.deu.edu.tr

---

[1] I pick up some of the topics from the following book:

- S. Theodoridis, and K. Koutroumbas (1999) *Pattern Recognition.* Academic Press.

especially the chapters that explain "*classifiers based on Bayes decision theory.*" The book is available, for example, at http://www.amazon.com. However, you do not need to buy since it's not necessarily easy to understand as it is. I pick up some of the topics and edited them in my way.

□ Contents:

1. What is *Pattern Recognition?*
   - ⋆ Hand-written character recognition as an example.
   - ⋆ Feature extraction.
2. *Principal Component Analysis (PCA):*

   How many features are enough?
3. *Sammon Mapping:*

   What a high-D feature space look like?
4. How can we observe multi-dimensional feature space?
5. To minimize classification error:

   Bayesian Classifier.
6. Feature Generation by *Linear Transforms.*

□ What is Pattern Recognition?

A typical example:

- Character recognition:

  characters as an example of patterns.

How can we recognize them?

- By taking all pixels as inputs?
- By some shape information?

  e.g.

  ∗ area within boundary;

  ∗ degree of how boundary curve.

  These are called *features*.

Feature space:

- l features forms *feature vector*

$$\mathbf{x} = (x_1, x_2, \cdots, x_l)$$

  for each object to be recognized.

- straight line separates the classes

  ⇓ in more general

  separation by *hyper-plane*

In short:

- pattern recognition is a *classifier*

□ Examples of feature to recognize characters:

- Perimeter:

$$P = \sum_{i=1}^{N-1} \|\mathbf{x}_{i+1} - \mathbf{x}_i\| + \|\mathbf{x}_N - \mathbf{x}_1\|. \tag{1}$$

- Area:

  Measure the area enclosed by a boundary by counting the number of pixels.

- Roundness ratio: $r = P^2/4\pi A$.
- Number of holes
- Bending energy:
  - defined at a point n as

$$E(n) = \frac{1}{P} \sum_{i=0}^{n-1} |k_i|^2 \tag{2}$$

  where k_i is the *curvature of the boundary* defined as:

$$k_n = \theta_{n+1} - \theta n, \quad n = 0, 1, \cdots, N-1. \tag{3}$$

  and

$$\theta_n = \tan^{-1} \frac{y_{n+1} - y_n}{x_{n+1} - x_n}, \quad n = 0, 1, \cdots, N-1. \tag{4}$$

  when we measure the length of the boundary at a point by the number of pixels n between this point and the origin of the curve.

- Number of corners:

  the number of points where the curvature k_i takes large values (infinity in theory)

□ Principal Component Analysis (PCA)

$\cdots$ A method of dimension reduction.

In the context of Pattern Recognition, a method to reduce the number of features without losing efficiency.

• Two dimensional cases:

Suppose we have N data pairs:

$$(x_i, y_i) \quad i = 1, 2, \cdots, N \tag{5}$$

In the context of Pattern Recognition, we have N data with two features.

Our objective is to find a and b that maximize the variance of the projections of these N points on the line

$$a(x - \bar{x}) + b(y - \bar{y}) = 0 \tag{6}$$

where $\bar{x}$ and $\bar{y}$ are the average of x_i and y_i, respectively.

Then

$$z_i = ax_i + by_i \tag{7}$$

is called *principal component* of x_i and y_i.[2]

In the context of Pattern Recognition, two features x_i and y_i are said to be reduced to one feature z_i.

[2] Yet another interpretation of z_i is a weighted sum of x_i and y_i. For example, x_i and y_i are point of homework assignment and final exam of student i and z_i is the evaluation.

Such (a, b) is known to be the *eigenvector*[3] of the matrix:

$$\begin{pmatrix} s_{xx} & s_{xy} \\ s_{yx} & s_{yy} \end{pmatrix} \tag{8}$$

where s_{xx} and s_{yy} are *variance* of x and y respectively and $s_{xy} = s_{yx}$ are *covariance* of x_i and y_i[4].

That is,

$$\lambda = \frac{s_{xx} + s_{xy} \pm \sqrt{s_{xx}^2 + s_{xy}^2 + s_{yy}^2 - s_{xx}s_{yy}}}{2} \tag{10}$$

and

$$(a, b) = (\frac{s_{xx} - s_{yy} \pm \sqrt{s_{xx}^2 + s_{xy}^2 + s_{yy}^2 - s_{xx}s_{yy}}}{2s_{xy}}, 1) \tag{11}$$

- Generalization to n-dimensional cases:

[3] If the transform of a vector $\mathbf{x}$ by a matrix A changes just a length of $\mathbf{x}$ by λ then $\mathbf{x}$ and λ are called *eigenvector* and *eigenvalue* of the matrix A.

[4] Variance of x_i and covariance of x_i and y_i are defined as

$$\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N} \quad \text{and} \quad \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N} \tag{9}$$

respectively.

□ **Sammon Mapping** (J. W. Sammon, 1969)

- Assume N points are given in the n-D space.

- Calculate distance matrix R ($N \times N$) whose i-j element is the Euclidean distance between the i-th and j-th point.

- Also think of a tentative N points in the 2-D space that are located at random at the beginning.

- The distance matrix Q is calculated in the same way as R.

- Then the error matrix $P = R - Q$ is defined.

- Search for the locations of N points in the 2-D space that minimizes the sum of element P.

This is an optimization problem which we can solve by GA, that is,

- N points in n-D space $\Rightarrow$ N points in 2-D space

  with the distance relation being preserved as much as possible; or

- The n-D distances are approximated by 2-D distances

  with a minimal error.

□ GA Implementation of Sammon Mapping

Chromosome:

(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	....	(x_N, y_N)

Recombination with Uniform Crossover:

(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	....	(x_N, y_N)
$\updownarrow$	$\updownarrow$	$\updownarrow$		$\updownarrow$
(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	....	(x_N, y_N)

Mutation:

□ Bayes Decision Theory

— for the design of classifiers.

Pattern Recognition is more or less statistical due to:

– statistic variation of patterns;

– statistic nature of feature selection;

– noise in employing sensors.

So, the task is:

– to classify unknown patterns into the *most probable class.*

Then

– what does "the most probable" means?

To answer this question:

– we study here the Bayesian Statistical Theory.

- When we classify an unknown pattern which is represented by a feature vector $\mathbf{x}$, we consider M conditional probabilities:
$$p(\omega_i|\mathbf{x}) \quad i = 1, \cdots, M \tag{12}$$

- Then we define "the most probable class" as the class ω_i which has the highest $p(\omega_i|\mathbf{x})$.

- We have the other possible statistical quantities:
$$p(\omega_i), \quad p(\mathbf{x}|\omega_i), \quad p(\mathbf{x})$$

- The *Bayesian rule* gives a relationship of these quantities:
$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{\sum_{k=1}^{M} p(\mathbf{x}|\omega_k)p(\omega_k)} \tag{13}$$
where the denominator is:
$$\sum_{k=1}^{M} p(\mathbf{x}|\omega_k)p(\omega_k) = p(\mathbf{x}). \tag{14}$$

- Example -1:
  — to understand the Bayesian Rule
  - We have two bags of no difference from its outlook.
  - One bag called R has 70 red balls and 30 blue balls.
  - The other bag called B has 30 red balls and 70 blue balls.
  - When we take one bag at random and pick up 12 balls, returning it to the bag at each time.
  - The result was 8 red balls and 4 blue one.
  - Then was the bag estimated to be R or B, and how probable the estimate is?

Clearly,

$$p(R) = p(B) = 1/2$$

$$\begin{aligned} p(D|R) &= {}_{12}C_8 (0.7)^8 (0.3)^4 \\ p(D|B) &= {}_{12}C_8 (0.3)^8 (0.7)^4 \end{aligned}$$

So we obtain

$$p(R|D) = \frac{(0.7)^8 (0.3)^4}{(0.7)^8 (0.3)^4 + (0.3)^8 (0.7)^4} \approx 0.97$$

(How big it is compared to our intuitive estimation!)

- Example -2:
  - Three prisoners (**A**, **B**, and **C**) are in a prison.
  - **A** knows the fact that the two out of the three are to be executed tomorrow, and the rest becomes free.
  - **A** thought either one of **B** or **C** is sure to be executed.
  - Then, **A** asked a guard "even if you tell me which of **B** and **C** is executed, that will not give me any information as for me. So please tell it to me."
  - The guard answers that **C** will. $\Rightarrow$ data D
  - Now, **A** knows one of **A** or **B** is sure to be free.

Do you guess probability $p(A|D) = 1/2$?

If this is correct, then the answer of the guard had given an information about **A**, since probability $p(A) = 1/3$.

Try to apply Bayesian rule, if you agree

$$p(A) = p(B) = p(C) = 1/3.$$

Then the conditional probabilities of the data, that is, "C will be executed" under the condition that A, B, and C will be free tomorrow, respectively, are

$$p(D|A) = 1/2 \tag{15}$$
$$p(D|B) = 1 \tag{16}$$
$$p(D|C) = 0 \tag{17}$$

- Now it's clear that
  - $p(A)$ and so on are to be called
    - ⋆ *a priori* probabilities;
  - and $p(A|D)$ and so on to be
    - ⋆ *a posteriori* probabilities.

- In the same way,
  - $p(\omega_i)$ is called
    - ⋆ *a priori* probability[5];
  - $p(\omega_i|\mathbf{x})$
    - ⋆ *a posteriori* probabilities.

Furthermore

- $p(\mathbf{x})$ is called
  - ⋆ p.d.f. of $\mathbf{x}$
- $p(\mathbf{x}|\omega_i)$
  - ⋆ class-conditional p.d.f.[6]
    - · which describes the distribution of the feature vectors $\mathbf{x}$ in each of the classes ω_i.

[5] Usually given, but if unknown, it can be estimated as N_i/N where N_i is the number of training samples which belong to class ω_i, and N is the total number of training samples

[6] This is also estimated from training data which will be explained later more in detail.

□ The Bayesian Classifier:

Definition 1 (The Bayesian Classifier) *classify* $\mathbf{x}$ *to the class* ω_i *such that* $p(\omega_i|\mathbf{x})$ *takes the maximum value.*

- Two-class case

$$\star \text{ if } p(\omega_1|\mathbf{x}) > p(\omega_2|\mathbf{x}), \quad \text{then classify } \mathbf{x} \textit{ to } \omega_1 \tag{18}$$

$$\star \text{ if } p(\omega_1|\mathbf{x}) < p(\omega_2|\mathbf{x}), \quad \text{then classify } \mathbf{x} \textit{ to } \omega_2 \tag{19}$$

· An example:

If we assume $p(\omega_1|\mathbf{x}) = N(0, 1/2)$, $p(\omega_2|\mathbf{x}) = N(1, 1/2)$[7], and $p(\omega_1) = p(\omega_2) = 1/2$, then how is our classification like?

The region where $p(\omega_1|\mathbf{x}) > p(\omega_2|\mathbf{x})$ is the region where

$$\frac{p(\mathbf{x}|\omega_1)p(\omega_1)}{p(\mathbf{x})} > \frac{p(\mathbf{x}|\omega_2)p(\omega_2)}{p(\mathbf{x})} \tag{20}$$

Since $p(\mathbf{x}) > 0$ and $p(\omega_1) = p(\omega_2)$, the region is where

$$p(\mathbf{x}|\omega_1) > p(\mathbf{x}|\omega_2) \tag{21}$$

holds. Now assume the two Gaussian distribution cross at $x = x_0$, we can conclude:

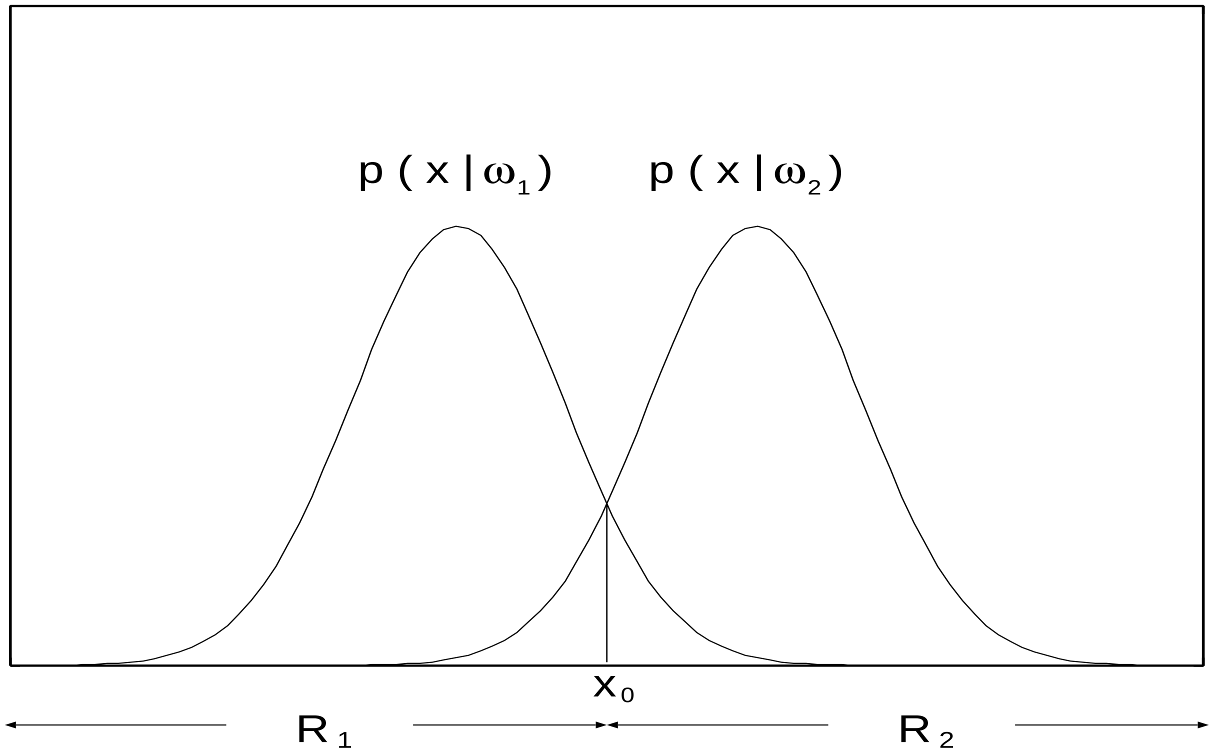
$\star$ if $x < x_0$ then classify $\mathbf{x}$ to class ω_1,
and vice versa.

---

[7] $N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{(x-\mu)^2}{2\sigma^2}$

□ Probability of Misclassification:

In the previous example:



- The probability of misclassifying $\mathbf{x}$ of ω_2 to ω_1 is

$$\int_{-\infty}^{x_0} p(\mathbf{x}|\omega_2)dx, \tag{22}$$

and the probability of misclassifying $\mathbf{x}$ of ω_1 to ω_2 is

$$\int_{x_0}^{\infty} p(\mathbf{x}|\omega_1)dx. \tag{23}$$

Therefore the total error probability is

$$\int_{-\infty}^{x_0} p(\mathbf{x}|\omega_2)dx + \int_{x_0}^{\infty} p(\mathbf{x}|\omega_1)dx. \tag{24}$$

- But usually feature space is multidimensional and so

$$\int_{R_1} p(\mathbf{x}|\omega_2)d\mathbf{x} + \int_{R_2} p(\mathbf{x}|\omega_1)d\mathbf{x} \tag{25}$$

is a more general expression of the equation (24).

- A generalization to M-class problem.
  - Assume that

$$R_j : \quad j = 1, 2, \cdots M. \tag{26}$$

    is the region of the feature space assigned to class ω_j.

  - Then the probability that a feature vector $\mathbf{x}$ that belongs to ω_k lies in R_i $(i \neq k)$ (i.e., misclassify $\mathbf{x}$ of ω_k to ω_i) is:

$$p_e = \sum_{i=1}^{M} \int_{R_i} p(\mathbf{x}|\omega_k) d\mathbf{x} \tag{27}$$

  - But usually, some errors are not so important as others, and λ_{ki}, importance of the error where $\mathbf{x}$ of ω_k to ω_i [8], is weighted as:

$$r_k = \sum_{i=1}^{M} \lambda_{ki} \cdot \int_{R_i} p(\mathbf{x}|\omega_k) d\mathbf{x} \;^{9} \tag{28}$$

    and the goal is to determine R_j such that

$$\begin{aligned} r &= \sum_{i=1}^{M} r_k \cdot p(\omega_k) \\ &= \sum_{i=1}^{M} \int_{R_i} \sum_{k=1}^{M} \lambda_{ki} \cdot p(\mathbf{x}|\omega_k) p(\omega_k) d\mathbf{x} \end{aligned} \tag{29}$$

    is minimized.

  - This is achieved if each of the integrals is minimized.

(to continue)

[8] In other words, λ_{ki} is a penalty term corresponding to a misclassification of $\mathbf{x}$ that belongs to ω_k lies in R_j

[9] Here, $\int_{R_i} p(\mathbf{x}|\omega_k) d\mathbf{x}$ is the probability of a feature vector from ω_k being classified in ω_i.

– Hence the following formula holds and called *Likelihood Ratio Test.*

Test 1 (The Likelihood Ratio Test) *Assign* $\mathbf{x}$ *to* R_i *if*

$$\sum_{k=1}^{M} \lambda_{ki} \cdot p(\mathbf{x}|\omega_k)p(\omega_k) < \sum_{k=1}^{M} \lambda_{kj} \cdot p(\mathbf{x}|\omega_k)p(\omega_k) \tag{30}$$

for $\forall j \neq k$*.*

– The two-class ($M = 2$) case:

· Assign $\mathbf{x}$ to ω_i if

$$\begin{aligned} & \lambda_{11}p(\mathbf{x}|\omega_1)p(\omega_1) + \lambda_{21}p(\mathbf{x}|\omega_2)p(\omega_2) \\ < \; & \lambda_{12}p(\mathbf{x}|\omega_1)p(\omega_1) + \lambda_{22}p(\mathbf{x}|\omega_2)p(\omega_2) \end{aligned} \tag{31}$$

That is,

$$(\lambda_{21} - \lambda_{22})p(\mathbf{x}|\omega_2)p(\omega_2) < (\lambda_{11} - \lambda_{11})p(\mathbf{x}|\omega_1)p(\omega_1) \tag{32}$$

If we assume $\lambda_{ij} > \lambda ii$,

· Assign $\mathbf{x}$ to ω_i if

$$\frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x}|\omega_1)} > \frac{p(\omega_2)}{p(\omega_1)} \cdot \frac{(\lambda_{21} - \lambda_{22})}{(\lambda_{12} - \lambda_{11})} \tag{33}$$

(to continue)

If two classes are equiprobable $p(\omega_1) = p(\omega_2)$ and $\lambda_{11} = \lambda_{22} = 0$, then

$$Assign\ \mathbf{x}\ to\ \omega_1\ if$$

$$p(\mathbf{x}|\omega_1) > p(\mathbf{x}|\omega_2) \cdot \frac{\lambda_{21}}{\lambda_{12}} \tag{34}$$

Furthermore if the importance of two misclassification is the same, that is, $\lambda_{12} = \lambda_{21} = 1$, then

$$p(\mathbf{x}|\omega_1) > p(\mathbf{x}|\omega_2) \tag{35}$$

This is equivalent of the equation (18).

□ Decision Surfaces and Discriminant Functions:

— to partition the feature space into M regions.

- If regions R_i and R_j are contiguous

$$p(\omega_i|\mathbf{x}) - p(\omega_j|\mathbf{x}) = 0 \tag{36}$$

determines the surface that partition R_i and R_j,

- Or, if necessary, using a monotonic function $f(x)$[10], we define:

$$g_i(x) = f(p(\omega_i|\mathbf{x})) \tag{37}$$

and we can say

$$\star\text{if } g_i(x) > g_j(x) \text{ for } \forall j \neq i, \text{classify } \mathbf{x} \text{ to } \omega_i. \tag{38}$$

Hence

$$g_{ij} \equiv g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0 \tag{39}$$

determines the *decision surfaces* separating contiguous regions and called *discriminant function.*

[10]e.g., $f(x) = \ln(x)$ for the Gaussian distribution.

□ Bayesian Classification for 2-d Gaussian pdf:

- l-dimensional Gaussian distribution

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{l/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu) \qquad (40)$$

where

$$\cdot \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1l} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2l} \\ & & \cdots & \\ \sigma_{l1} & \sigma_{l2} & \cdots & \sigma_l^2 \end{pmatrix} \qquad (41)$$

- $\cdot\ \mathbf{x} = (x_1, x_2, \cdots, x_l)$;
- $\cdot\ \mu$ is a l-dimensional vector whose i-th element μ_i is the mean value of x_i;
- $\cdot\ \sigma_i^2$ is the variance of x_i;
- $\cdot\ \sigma_{ij}$ is the covariance between x_i and x_j;
- $\cdot\ |\Sigma|$ is determinant of the matrix Σ;
- $\cdot\ \Sigma^{-1}$ is inverse matrix of Σ[11];
- $\cdot\ \mathbf{x}^T$ is transverse of vector $\mathbf{x}$[12].

---

[11] If we assume

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad (42)$$

then

$$|A| = ad - bc \qquad (43)$$

and

$$A^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \qquad (44)$$

[12] That is, when $\mathbf{x} = (x_1, x_2)$

$$\mathbf{x}^T = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \qquad (45)$$

• 2-dimensional Gaussian

We can calculate $l = 2$ version from the equation (40) as:

$$p(x_1\ ,\ x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \cdot \exp\{(-\frac{1}{2(1-\rho^2)} \cdot ((\frac{x_1-\mu_1}{\sigma_1})^2 - 2\rho(\frac{x_1-\mu_1}{\sigma_1})(\frac{x_2-\mu_2}{\sigma_2}) + (\frac{x_2-\mu_2}{\sigma_2})^2)\} \tag{46}$$

where ρ is correlation coefficient defined as:

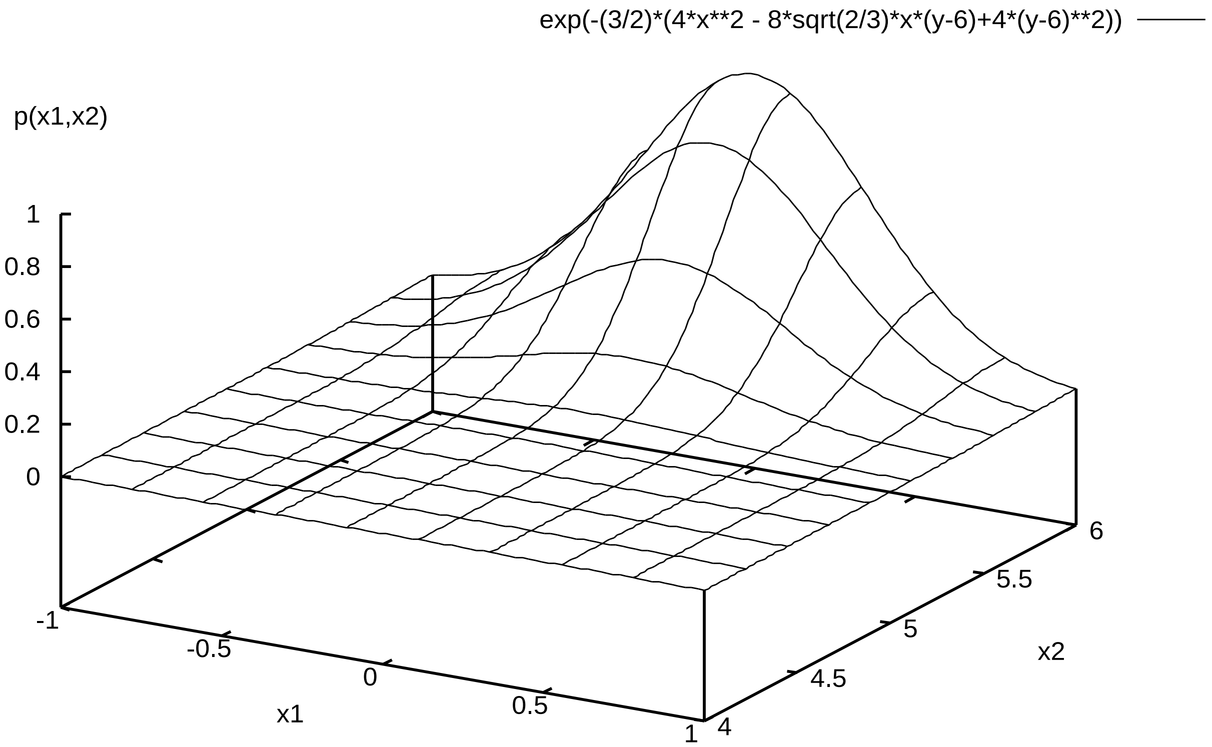
$$\rho = \frac{\sigma_{12}}{\sigma_1\sigma_2} \tag{47}$$

Exercise 1:

1. Derive the equation (46) from the equation (40).
2. By giving your own set of five parameters:

   μ_1, μ_2, σ_1, σ_2, and σ_{12}

   Draw the Gaussian surface on (x_1, x_2) coordinate. Then explore a couple of configurations of these five parameters.


exp(-(3/2)*(4*x**2 - 8*sqrt(2/3)*x*(y-6)+4*(y-6)**2))
p(x1,x2)
1
0.8
0.6
0.4
0.2
0
-1
-0.5
0
0.5
1
x1
4
4.5
5
5.5
6
x2

□ Examples of Decision Surface in 2D Feature Space:

The equation (46) with taking the function *ln* as $f(\cdot)$ in the equation (37), we obtain our discriminating function g_i as:

$$\begin{aligned} g_i(\mathbf{x}) &= \ln(p(\mathbf{x}|\omega_i)p(\omega_i)) \\ &= \ln p(\mathbf{x}|\omega_i) + \ln p(\omega_i) \\ &= -\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i) + \ln p(\omega_i) + c_i \\ &= -\frac{1}{2}\mathbf{x}^T\Sigma_i^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i \\ &\qquad + \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mathbf{x} + \ln p(\omega_i) + c_i \end{aligned} \tag{48}$$

where

$$c_i = -(l/2)\ln(2\pi) - (1/2)\ln|\Sigma_i| \tag{49}$$

• Example 1

When

$$\Sigma_i = \begin{pmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{pmatrix}$$

$$g_i = \frac{x_1^2 + x_2^2}{-2\sigma_i^2} + \frac{\mu_{i1}x_1 + \mu_{i2}x_2}{\sigma_i^2} - \frac{\mu_{i1}^2 + \mu_{i2}^2}{-2\sigma_i^2} + \ln p(\omega_i) + c_i \quad (50)$$

⋆ Hence, under this kind of two distributions, the decision surface is a *circle*.

• Example 2

When

$$\Sigma_i = \begin{pmatrix} \sigma_{i1}^2 & 0 \\ 0 & \sigma_{i2}^2 \end{pmatrix}$$

$$g_i = \frac{\sigma_{i2}^2 x_1^2 + \sigma_{i1}^2 x_2^2}{-2\sigma_{i1}^2\sigma_{i2}^2} + \frac{\sigma_{i2}^2 \mu_{i1} x_1 + \sigma_{i1}^2 \mu_{i2} x_2}{\sigma_{i1}^2 \sigma_{i2}^2} - \frac{\sigma_{i2}^2 \mu_{i1}^2 + \sigma_{i1}^2 \mu_{i2}^2}{-2\sigma_{i1}^2\sigma_{i2}^2} + \ln p(\omega_i) + c_i \quad (51)$$

⋆ This kind of two distributions construct a *quadratic* decision surface.

⋆ If expanded to $l > 2$ case, then it is called *hyper-quadratic*.

• Exercise 2:

When $p(\omega_1) = p(\omega_2)$, $\mu_1 = (0, 0)$, and $\mu_2 = (1, 0)$, obtain decision surface in the following two cases.

(1) $\Sigma_1 = \begin{pmatrix} 0.10 & 0 \\ 0 & 0.15 \end{pmatrix}, \qquad \Sigma_2 = \begin{pmatrix} 0.20 & 0 \\ 0 & 0.25 \end{pmatrix}$

(2) $\Sigma_1 = \begin{pmatrix} 0.10 & 0 \\ 0 & 0.15 \end{pmatrix}, \qquad \Sigma_2 = \begin{pmatrix} 0.15 & 0 \\ 0 & 0.10 \end{pmatrix}$

□ An expansion to l-D Feature space:

Our discriminant function now is the equation (48), i.e.,

$$g_i = -\frac{1}{2}\mathbf{x}^T\Sigma_i^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\Sigma_i^{-1}\mu_i - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mathbf{x} + \ln p(\omega_i) + c_i \tag{52}$$

We have discussed that this gives a *quadratic* decision surface.

But if we assume here:

$$\forall\Sigma_i = \Sigma \tag{53}$$

That is, all the variance matrices are identical, then the first and the last term do not contribute to $g_i - g_j$, and we can re-define g_i as:

$$g_i = \frac{1}{2}\mathbf{x}^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mathbf{x} - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i + \ln p(\omega_i) \tag{54}$$

And we can simplify the equation (54) as:[13]

$$g_i = w_i^T \mathbf{x} + w_{i0} \tag{55}$$

by putting

$$w_i = \Sigma^{-1} \mu_i \tag{56}$$

and

$$w_{i0} = \ln p(\omega_i) - \frac{1}{2} \mu_i^T \Sigma_i^{-1} \mu_i \tag{57}$$

Hence our new g_i is a linear function of $\mathbf{x}$ and the decision surfaces determined by $g_i - g_j = 0$ are *hyper-planes* instead of the previous *quadratic* surfaces.

[13]Since $w_i^T = \Sigma^{-1}\mu_i$ and Σ is symmetric, we obtain $w_i^T = \mu_i^T(\Sigma^{-1})^T = \mu_i^T(\Sigma^T)^{-1} = \mu_i^T\Sigma^{-1}$. And we conclude:

$$\frac{1}{2}\mathbf{x}^T w_i + \frac{1}{2} w_i^T \mathbf{x} = w_i^T \mathbf{x}.$$

Note that $\mathbf{x}^T$ is a raw vector and w_i is a column vector. Hence, $\mathbf{x}^T w_i = w_i^T \mathbf{x}$ holds.

□ Further two assumptions for simplification:

- When $\Sigma = \sigma^2 I$:

$$
\begin{aligned}
g_i - g_j &= (\frac{1}{\sigma^2}\mu_i^T \mathbf{x} + w_{i0}) - (\frac{1}{\sigma^2}\mu_j^T \mathbf{x} + w_{j0}) \\
&= \frac{1}{\sigma^2}(\mu_i^T - \mu_j^T)\mathbf{x} + w_{i0} - w_{j0} \qquad (58)
\end{aligned}
$$

Here, we put

$$w^T = \mu_i^T - \mu_i^T \qquad (59)$$

Then,

$$
\begin{aligned}
\sigma^2(g_i - g_j) &= w^T\mathbf{x} + \sigma^2(w_{i0} - w_{j0}) \\
&= \cdots \text{(see next page!)} \\
&= w^T(\mathbf{x} - \mathbf{x}_0) \qquad (60)
\end{aligned}
$$

where

$$x_0 = -\frac{1}{2}(\mu_i + \mu_j) - \sigma^2 \cdot \ln\{\frac{p(\omega_i)}{p(\omega_i)}\} \cdot \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2} \qquad (61)$$

In equation (60)

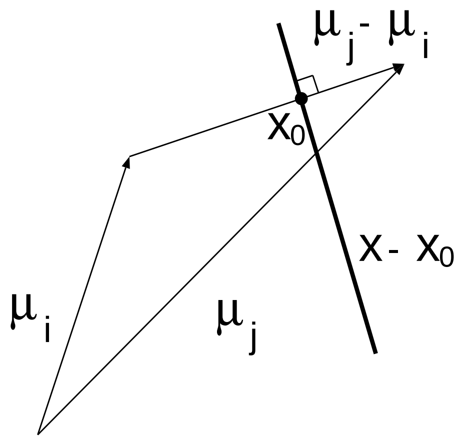
$$
\begin{aligned}
\sigma^2(w_{i0} - w_{i0}) &= \sigma^2\Big(\ln p(\omega_i) - \frac{1}{2}\frac{1}{\sigma^2}\mu_i^T\mu_i\Big) - \sigma^2\Big(\ln p(\omega_j) - \frac{1}{2}\frac{1}{\sigma^2}\mu_j^T\mu_j\Big) \\
&= \sigma^2\Big((\ln p(\omega_i) - \ln p(\omega_j)) - \frac{1}{2}(\mu_i^T\mu_i - \mu_j^T\mu_j)\Big) \\
&= \sigma^2 \ln\frac{p(\omega_i)}{p(\omega_j)} - \frac{1}{2}(\|\mu_i\|^2 - \|\mu_j\|^2) \\
&= (\mu_i - \mu_j)^T - \sigma^2 \cdot \ln\frac{p(\omega_i)}{p(\omega_j)} \cdot \frac{(\mu_i - \mu_j)^T(\mu_i - \mu_j)}{(\mu_i - \mu_j)^T(\mu_i - \mu_j)} \\
&= (\mu_i - \mu_j)^T\Big(-\frac{1}{2}(\mu_i + \mu_j) - \sigma^2 \cdot \ln\frac{p(\omega_i)}{p(\omega_j)} \cdot \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2}\Big) \\
&\equiv w^T\mathbf{x}_0
\end{aligned} \tag{62}
$$

Hence, from

$$\sigma^2(g_i - g_j) = w^T(\mathbf{x} - \mathbf{x}_0) = 0 \tag{63}$$

we can conclude:

$$\sigma^2(\mu_i - \mu_j)^T(\mathbf{x} - \mathbf{x}_0) = 0 \tag{64}$$



- The point $\mathbf{x}_0$ is on the vector $(\mu_i - \mu_j)$ and the location depends on σ^2 and $p(\omega_i)/p(\omega_j)$, that is,

  · If $p(\omega_i) > p(\omega_j)$, then the hyper-plane is located closer to μ_i.

  · If σ^2 is small w.r.t. $\|\mu_i - \mu_j\|$, then the location of the hyper-plane is insensitive.

- Non-diagonal Covariance Matrix Σ

  - As in the previous assumption we obtain:

$$g_i - g_j = w^T(\mathbf{x} - \mathbf{x}_0) \tag{65}$$

where

$$w^T = \mu_i^T - \mu_i^T \tag{66}$$

and

$$x_0 = -\frac{1}{2}(\mu_i + \mu_j) - \sigma^2 \cdot \ln\{\frac{p(\omega_i)}{p(\omega_i)}\} \cdot \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2_{\Sigma^{-1}}} \tag{67}$$

  - Here, we refer $\|\mathbf{x}\|_{\Sigma^{-1}} \equiv (\mathbf{x}^T\Sigma^{-1}\mathbf{x})$ to Σ^{-1} *norm* of $\mathbf{x}$.

  - The decision hyper-plane is no longer orthogonal to $\mu_i - \mu_j$ but to its linear transformation $\Sigma^{-1}(\mu_i - \mu_j)$.

  · Under *non-diagonal* Σ, the Σ^{-1} norm

□ Classification by the distance from the mean:

Recall that we classify $\mathbf{x}$ to ω_i such that g_i is maximized,

$\Downarrow$ then it implies

- for diagonal $\Sigma = \sigma^2 I$, minimize

$$d_E \equiv \|\mathbf{x} - \mu_i\| \tag{68}$$

- for no-diagonal Σ, minimize

$$d_M \equiv (\mathbf{x} - \mu_i)^T \Sigma^{-1} (\mathbf{x} - \mu_i) \tag{69}$$

which is called *Mahalanobis distance.*

Note that:

- $d_E = const. \Rightarrow$ *hyper-sphere*
- $d_M = const. \Rightarrow$ *hyper-ellipse*

□ Diagonalization of Variance Matrix:

— to learn the shape of the ellipses

Σ can be diagonalized by a *unitary transform* Φ as:

$$\Sigma = \Phi^{-1}\Lambda\Phi. \tag{70}$$

where $\Phi^T = \Phi^{-1}$ because Φ is unitary, and Λ is the diagonal matrix whose elements are the eigenvalues of Σ. Φ has eigenvectors of Σ as its column, that is,

$$\Phi = (\nu_1 \nu_2 \cdots \nu_l). \tag{71}$$

From (69) and (70), our hyper-ellipse becomes

$$(\mathbf{x} - \mu_i)^T \Phi^{-1}\Lambda\Phi(\mathbf{x} - \mu_i) = const. \tag{72}$$

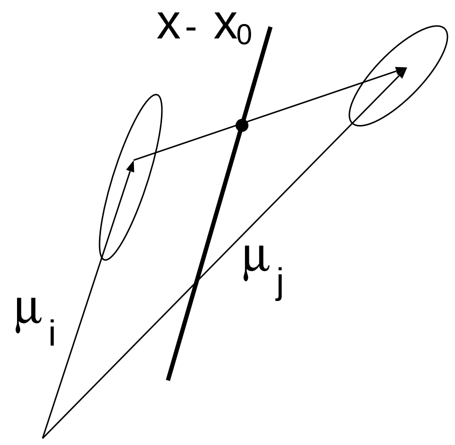
If we define $\mathbf{x}' = \Phi\mathbf{x}$ here,

$$\mathbf{x}' = \nu_k^T \mathbf{x} \tag{73}$$

implies the *projections* of $\mathbf{x}$ onto each eigenvector, and (72) becomes our familiar hyper-ellipse formula:

$$\frac{(x_1' - \mu_{i1}')}{\lambda_1} + \frac{(x_2' - \mu_{i2}')}{\lambda_2} + \cdots + \frac{(x_l' - \mu_{il}')}{\lambda_l} = \text{const.} \tag{74}$$

Thus, all points having the same Mahalanobis distance from a specific point are located on an ellipse.



• Exercise 3:

About a two-class, two dimensional Bayesian classifier where $p(\omega_1) = p(\omega_2)$, $\mu_1 = (0,0)$, $\mu_2 = (1,0)$, and

$$\Sigma_1 = \Sigma_2 = \Sigma = \begin{pmatrix} 1.1 & 0.3 \\ 0.3 & 1.9 \end{pmatrix}$$

calculate the followings.

(1) Obtain d_M of $\mathbf{x} = (1.0, 2.2)$ from μ_1 and μ_2, respectively.

(2) Should $\mathbf{x}$ classified to ω_1 or ω_2?
(Compare the result when we use d_E instead of d_M.

(3) Obtain the ellipse when we suppose *const.* $= 1$.

$$(\mathbf{x} - \mu_i)^T \Sigma^{-1} (\mathbf{x} - \mu_i) = 1, \quad i = 1, 2.$$

(4) Calculate eigenvalues λ_1 and λ_2, and eigenvectors ν_1 and ν_2 of Σ.

(5) Obtain Φ and Λ and ascertain $\Sigma = \Phi^{-1}\Lambda\Phi$.

(6) Obtain the projected ellipse

$$(\mathbf{x} - \mu_i)^T \Phi^{-1} \Lambda \Phi (\mathbf{x} - \mu_i) = 1, \quad i = 1, 2.$$

□ What to do with unknow pdf?

So far we have studied known pdf $p(\mathbf{x}|\omega_i)$ with its known parameter which is not of usual case.

Usually,

- Pdf is known but some parameters are unknown; or
- underlying pdf should be estimated from the available data.

□ Non-parametric pdf Estimation

— Histogram approximation of unknown pdf.

- 2-D case:

(1) x-axis is divided into successive *bins* of length h.

(2) If N is the total number of samples and k_N of them are located inside a bin then pdf is approximated by

$$p(x) \equiv p(\hat{x}) \approx \frac{1}{h}\frac{k_N}{N} \tag{75}$$

for

$$|x - \hat{x}| \leq \frac{h}{2} \tag{76}$$

- l-D case:

In the same way as 2-D case, l-dimensional space is divided into hypercubes with length of side h whose volume is h^l.

- Assume we have N feature vectors:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{il}), \quad i = 1, 2, \cdots, N. \tag{77}$$

and a function $\phi(\mathbf{x}_i)$:

$$\phi(\mathbf{x}_i) = \begin{cases} 1 & \text{for} \quad |x_{ij}| \leq 1/2 \\ 0 & \text{otherwise.} \end{cases} \tag{78}$$

This is referred to as *Parzen Windows*

(to continue)

Then

$$p(\mathbf{x}) \equiv p(\mathbf{x}_i) \approx \frac{1}{h^l}\left(\frac{1}{N}\sum_{i=1}^{N}\phi\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)\right) \tag{79}$$

in which $\Sigma\,\phi(\cdot)$ implies the number of points falling inside the hypercube with length of side h centered at $\mathbf{x}_i$, the point where the pdf is to be estimated, that is to say, $\Sigma\,\phi = k_N$. In short:

$$p(\mathbf{x}) \approx \frac{1}{h^l}\left(\frac{k_N}{N}\right) \tag{80}$$

- How to determine h and N.

h: starting with a certain h, modify it iteratively to minimize misclassification error. (typically ranging from 0.1 to 0.8)

N: If N is appropriate for the 2-D version, the n-D version needs N^n samples.

- The likelihood ratio test using the Parzen Windows:

Assign $\mathbf{x}$ *to* ω_1 *if*

$$\frac{\frac{1}{N_1 h^l}\sum_{i=1}^{N_1}\phi\left(\frac{\mathbf{x}_1 - \mathbf{x}}{h}\right)}{\frac{1}{N_2 h^l}\sum_{i=1}^{N_2}\phi\left(\frac{\mathbf{x}_2 - \mathbf{x}}{h}\right)} > \frac{p(\omega_2)}{p(\omega_1)} \cdot \frac{(\lambda_{21} - \lambda_{22})}{(\lambda_{12} - \lambda_{11})} \tag{81}$$

□ k Nearest Neighbor Density Estimation (kNN)

While in the Parzen windows h is fixed, here in kNN it is k_N that is fixed.

- k Nearest Neighbor pdf:

  When the volume of the hypercube[14] around the point $\mathbf{x}$, which includes $k_N = k$ (fixed) out of a total of N samples, is $V(\mathbf{x})$ then the pdf is estimated as:

$$p(\mathbf{x}) = \frac{k}{NV(\mathbf{x})} \tag{82}$$

  Note that the lower the density of samples the larger the volume V , and vice versa.

- The likelihood ratio test:

  Assign $\mathbf{x}$ *to* ω_1 *if*

$$\frac{N_2 V_2}{N_1 V_1} > \frac{p(\omega_2)}{p(\omega_1)} \cdot \frac{(\lambda_{21} - \lambda_{22})}{(\lambda_{12} - \lambda_{11})} \tag{83}$$

[14] Instead of hypercube, we might compute its Euclidean distance from all the training vectors of the different classes, that is, compute radius of the hypersphere centered at $\mathbf{x}$ that contains k points from ω_i. Then $V(\mathbf{x})$ is the volume of the hypersphere.

□ The Nearest Neighbor Density Estimation (k=1)

Algorithm 1 (The k Nearest Neighbor) *Given an unknown feature vector* $\mathbf{x}$ *and a distance measure like Euclidean or Mahalanobis distance:*

1. *Out of the* N *training vectors, identify the* k *(an odd integer) nearest neighbors, irrespective of class.*
2. *Out of these* k *samples, identify the number of vectors* k_i *that belong to class* ω_i $i = 1, 2, \cdots M$. $(\sum k_i = k)$.
3. *Assign* $\mathbf{x}$ *to* ω_i *with the maximum* k_i.

- The simplest version is when $k = 1$. This called Nearest Neighbor Algorithm (NN). Then:

  $\mathbf{x}$ *is assigned to its nearest neighbor class.*

- kNN is no more the Bayesian classification but

$$P_B \leq P_{NN} \leq P_B(2 - \frac{M}{M-1}P_B) < 2P_B \qquad (84)$$

$$P_B < P_{kNN} \leq P_B + \sqrt{\frac{2P_{NN}}{k}} \qquad (85)$$

  are known, where P_B and P_{kNN} is misclassification error probabilities of Bayesian and kNN classifier, respectively.

(to continue)

And when $P_B \ll 1$, we obtain that

$$P_{NN} \approx 2P_B \tag{86}$$

and

$$P_{3NN} \approx P_B + 3P_B^2 \tag{87}$$

When $P_B = 0.01$, for example,

$$P_{3NN} - P_B \approx 0.00003 \tag{88}$$

Further, as k becomes large, kNN tends to the Bayesian classifier.

Definition 1 (Voronoi Tesselation) *When $k = 1$, N training vectors $\mathbf{x}_i$ $(i = 1, 2, \cdots, N)$ define a partition of the l-D feature space into N regions:*

$$R_i = \{\mathbf{x} | d(\mathbf{x}, \mathbf{x}_i) < d(\mathbf{x}, \mathbf{x}_j)\} \quad i \neq j. \tag{89}$$

R_i contains all points that are closer to $\mathbf{x}_i$ than any other point.

□ Thought Experiments

- You have (one dimensional) 100 positive valued training samples created following[15]:

  $$p(\mathbf{x}|\omega_1) = x\exp(-x^2/2) \tag{91}$$

  and the other 100 following

  $$p(\mathbf{x}|\omega_2) = 2x\exp(-x^2) \tag{92}$$

  but you don't know both of the function from which data are created.

  (1) Classify a feature vector $\mathbf{x}$ produced at random.

  (2) For each set of 100 data, observe the histogram with various value of h.

  (3) Observe the histograms by changing N_1 and N_2 from 10 to 1000.

[15] This is the Rayleigh pdf defined as

$$p(x) = \frac{x}{\sigma^2}\exp(-\frac{x^2}{2\sigma^2}.) \tag{90}$$

- For two sets of (2-dimensional) 100 data each drawn from

$$p(\mathbf{x}|\omega_1) = \frac{1}{2\pi}\exp(-(\mathbf{x}-\mu_1)^T(\mathbf{x}-\mu_1)) \tag{93}$$

$$p(\mathbf{x}|\omega_2) = \frac{1}{2\pi}\exp(-(\mathbf{x}-\mu_2)^T(\mathbf{x}-\mu_2)) \tag{94}$$

  where $\mu_1 = (1,1)$ and $\mu_2 = (3,3)$.

  (1) Classify a randomly cleated feature vector $\mathbf{x}$ using the Parzen windows.

  (3) Try 3NN Algorithm for
    i) hypercube of length h;
    ii) hypersphere of radius r
    iii) a hyper ellipsoid

  (3) Each of the above three cases, try to use likelihood ratio test.

- Create 100 training data following 3-dimensional Gaussian distribution with

$$\Sigma = \begin{pmatrix} 0.3 & 0.1 & 0.1 \\ 0.1 & 0.3 & -0.1 \\ 0.1 & -0.1 & 0.3 \end{pmatrix} \tag{95}$$

  where $\mu_1 = (0,0,0)$ and $\mu_2 = (1/2,1/2,1/2)$.

  (1) Classify a randomly cleated feature vector $\mathbf{x}$ using
    i) the Parzen windows;
    ii) 3NN Algorithm
    iii) NN Algorithm

  (2) Try to use likelihood ratio test.

□ Parameter Estimation under Known pdf

- Assume that we have 100 data:

$$3.2, 2, 8, 5.4, 6.2 \cdots \tag{96}$$

  that are drawn from $p(\mathbf{x}|\omega_i)$, a pdf for a class i.

- Sometimes
  · We don't have no information as for pdf, which we have studied in the previous section.
  or
  · We know the function form but not parameters, which we are going to study in this section.

• Maximum Likelihood Estimation:

- For a data set

$$\mathbf{x} = (x_1, x_2, \cdots, x_N),$$

  when likelihood that $\mathbf{x}$ is drawn from the pdf with parameter θ is expressed as $p(\mathbf{x};\theta)$, we might say "$p(\mathbf{x};\theta_1)$ is more likely than $p(\mathbf{x};\theta_2)$."

- Then which θ gives most likely $p(\mathbf{x};\theta)$?

(to continue)

- Under an assumption of independence between the different sample,

$$p(\mathbf{x};\theta) \equiv p(x_1, \cdots, x_N) = \prod_{k=1}^{N} p(\mathbf{x};\theta) \tag{97}$$

- To answer the question, likelihood function is defined as

$$l(\theta) = \prod_{k=1}^{N} p(\mathbf{x}_k;\theta) \tag{98}$$

- The necessary condition of the most likely $\theta = \theta_{ML}$ is

$$\frac{\partial \prod_{k=1}^{N} p(\mathbf{x}_k;\theta)}{\partial \theta} = 0 \tag{99}$$

- Or sometimes more conveniently taking logarithm

$$L(\theta) = \ln \prod_{k=1}^{N} p(\mathbf{x}_k;\theta) \tag{100}$$

called *loglikelihood function*,

- and θ_{ML} is obtained from

$$\frac{\partial L}{\partial \theta} = \sum \frac{1}{p(\mathbf{x}_k;\theta)} \cdot \frac{\partial p(\mathbf{x}_k;\theta)}{\partial \theta} \tag{101}$$

Exercise

1. We now have 100 positive values

$$2.4, 5.6, 3.2 \cdots$$

and we know they are drawn from

$$p(\mathbf{x};\theta) = \theta^2 x \exp(-\theta x) \tag{102}$$

called *Erlang* pdf, but we don't know the value of θ. Then what θ is the most likely value for the given 100 data?

2. We have 100 data drawn from $N(\mu, 1)$. What is the most likely estimate for μ?

- Maximum a posteriori pdf estimation
  - For a data set $X = (\mathbf{x}_1, \cdots, \mathbf{x}_N)$ θ, which is thought as a random variable here, is estimated its value on the condition that the data $\mathbf{X}$ has occurred.

    Since

$$p(\theta|X) = \frac{p(\theta)p(X|\theta)}{p(X)} \tag{103}$$

    θ_{MAP} is estimated from

$$\frac{\partial}{\partial\theta}\, p(\theta|X) = \frac{\partial}{\partial\theta}\, p(\theta)p(\theta|X) = 0 \tag{104}$$

  - The difference between ML and MAP is that the latter involves $p(\theta)$. Therefore if $p(\theta)$ is uniform distribution or if $p(\theta)$ has a large variance, the Both are the same or almost the same. But in general the two estimation yield a different results.

(to continue)

Example:

Assume we have 100 data from Gaussian pdf with unknown μ and we know μ is from Rayleigh pdf:

$$p(\mu) = \frac{\mu \exp(-\mu^2/2\sigma_\mu^2)}{\sigma_\mu^2} \tag{105}$$

Estimate μ_{MAP}.[16]

Yet Another Example:

Assume $x_1, x_2, \cdots, x_N$ are data from Gaussian pdf with known σ and unknown μ, that is,

$$p(x_k;\mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(x_k-\mu)^2}{2\sigma^2}\} \tag{108}$$

and μ is known to be Gaussian distributed around the value μ_0, that is,

$$p(\mu) = \frac{1}{\sqrt{2\pi}\sigma_\mu} \exp\{-\frac{(\mu-\mu_0)^2}{2\sigma_{\mu^2}}\} \tag{109}$$

Estimate μ_{MAP}.[17]

---

[16]The answer is

$$\mu_{\mathrm{MAP}} = \frac{Z}{2R}(1+\sqrt{1+\frac{4R}{Z^2}}). \tag{106}$$

where

$$Z = \frac{1}{\sigma^2}\sum_{k=1}^{N} x_k, \quad \text{and} \quad R = \frac{N}{\sigma^2} + \frac{1}{\sigma_\mu^2}. \tag{107}$$

[17]The answer is

$$\mu_{\mathrm{MAP}} = \frac{\mu_0 + (\sigma_\mu^2/\sigma^2)\sum_{k=1}^{N} x_k}{1+(\sigma_\mu^2/\sigma^2)N} \tag{110}$$

which can be interpreted as $\mu_{\mathrm{MAP}} = \mu_0$ when $\sigma_\mu^2/\sigma^2 \gg 1$. I thank Lalehan for pointing this out!

□ Unknown pdf shortly revisited

- Maximum Entropy Estimation:
  - Determine $p(x)$ such that entropy

$$H = -\int_x p(x) \ln p(x) dx \tag{111}$$

    is maximized.

- Estimation as a sum of known pdf:
  - Assuming each sample x is drawn from any of the J pdfs with the probability p_j.

$$p(x) = \sum_{j=1}^{J} p(x|j) p_j \tag{112}$$

⋆ This concludes the course of this semister.
⋆ Thank you very much for your interest and atendance.
⋆ See you.