Introduction to Pattern Recognition

Dokuz Eylül University
Computer Engineering Department

(25 September, 2000 – 5 January, 2001)

□ Contents:

1 What is Pattern Recognition?

2 What for?

A typical example:

- character recognition

  characters as an example of patterns

But how?

- 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

$$\Downarrow \text{ in more general}$$

  separation by *hyper-plane*

In short:

- pattern recognition is a *classifier*

Yet another example

- machine vision (robot eye);
- face recognition;
- speech recognition;
- computer-aided diagnosis;
- what else?

Problems:

- What features are to be used to classify?
- What is the optimum number of features?
- How is the classifier designed?

Keywords:

- **features, feature-space, feature-vectors, hyperplane, classifiers,**

3 supervised/unsupervised

Supervised pattern recognition

- classifies unknown data using a set of training data.
- Learning with teacher.

Unsupervised pattern recognition (clustering)

- unravel the underlying similarities and cluster similar vectors together using a set of feature vectors

  e.g.

  - Assume we are interested in studying relations among a country's GNP, the level of people's illiteracy, and a children's mortality rate.

    $$\Downarrow$$

    three-dimensional feature vectors
- Learning without teacher.

4 Hand-written character recognition – revisit

What should be employed as features:

1. Pixels;
2. Geometric Features;
3. Fourier Features;
4. Chain Codes;
5. Moment-based Features;
6. Neural Networks.

Verify each technique by your own several handwriting figures, say, 1's, 5's, and 8's, on a scale paper (with each character being on 1cm x 1cm = 100 pixels section for instance).

1. Each pixel's state as a set of features:

- draw 20 different writings for each character;
- keep the pixel data for the future reference.

2. Geometric Features:

- 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

$$r = \frac{P^2}{4\pi A} \tag{2}$$

- Bending energy;
  (later in detail)

- The number of corners in the boundary;
  (later in detail)

- The number of holes.

3. Fourier Features:

Let

$$(x_k, y_k), \quad k = 0, 1, \cdots, N-1. \tag{3}$$

be the coordinates of N samples on the boundary of an image region. For each pair (x_k, y_k) we define the complex:

$$u_k = x_k + jy_k. \tag{4}$$

For the N u_k points we obtain the DFT f_l

$$f_l = \sum_{k=0}^{N-1} u_k \exp(-j2\pi lk/N), \quad l = 0, 1, \cdots, N-1. \tag{5}$$

and call them l Fourier Descriptors of the boundary.

Another way is from curvature function of the boundary:

$$k(t) = \frac{d\theta(t)}{dt}. \tag{6}$$

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, the curvature of the boundary is approximated by

$$\theta_n = \tan^{-1} \frac{y_{n+1} - y_n}{x_{n+1} - x_n}, \quad n = 0, 1, \cdots, N-1. \tag{7}$$

$$k_n = \theta_{n+1} - \theta n, \quad n = 0, 1, \cdots, N-1. \tag{8}$$

Bending energy and number of corners revisited:

Then we can define

- bending energy at a point n as

$$E(n) = \frac{1}{P}\sum_{i=0}^{n-1} |k_i|^2 \tag{9}$$

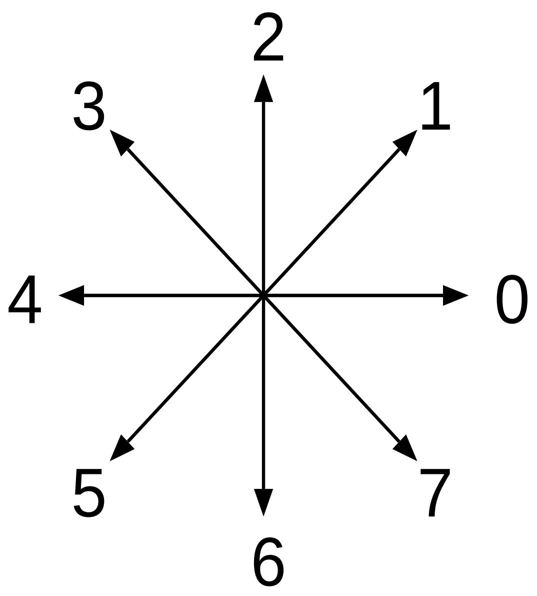
and

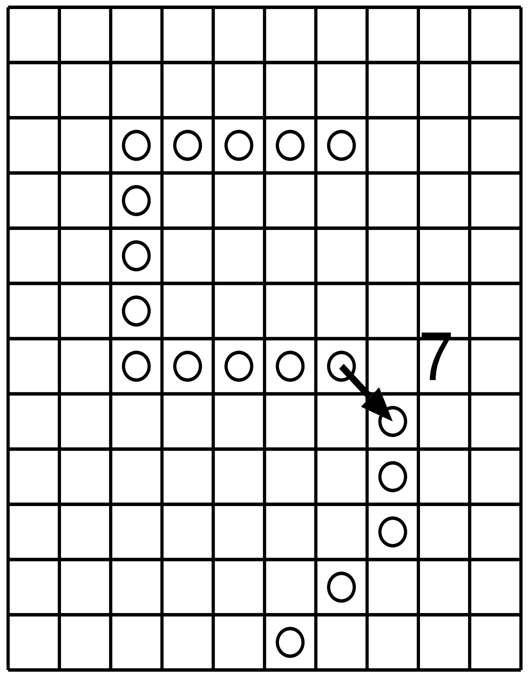
- number of corners in the boundary contour as
  - the number of points where the curvature k_i takes large values (infinity in theory)

4. Chain codes:

□ Chain code $(d_1, d_2, \cdots, d_N)$ is defined as:

a series of direction from pixel (x_i, y_i) to (x_{i+1}, y_{i+1}) on the boundary with, say, 8 possible directions.





- Direction and Direction-length Features:

$$\frac{\text{the number of each direction}}{\text{the total number of chain codes}} \tag{10}$$

or

$$\frac{\text{the total length of the line segments in each direction}}{\text{the total length of the boundary line}} \tag{11}$$

$\Rightarrow$ 8 features

- Curvature Features:
  - Is an external angle between adjacent two edges at a corner of the polygon that is formed by the line segments
    - concave (smaller than 180°) or
    - convex (larger than 180°)?

    when the boundary is scanned in the clockwise sense.
  - 48 possible cases
    - Concave cases:
      - 01 02 03 12 13 14 23 24 25 34 35 36
        45 56 47 50 56 57 60 61 67 70 71 72
    - Convex cases:
      - 05 06 07 10 16 17 20 21 27 30 31 32
        41 42 43 52 53 54 63 64 65 74 75 76
  - Occurrence percentage of each of these cases.

$\Rightarrow$ 48 features

□ Summary:

What features can be used to recognize hand-written characters?

- Boundary-based:
  - Single feature:
    * Perimeter of the boundary;;
    * Area inside the boundary;
    * Roundness ratio;
    * Number of corners in the boundary;
    * Number of holes.
  - Multiple features:
    * Fourier features;
    * Chain codes.
- Pixel-based:
  - Neural-networks
    * Perceptron learning
    * Genetic algorithm

Genetic Algorithm (GA)

□ An example:

- Neural network with 100 input neurons and 10 output neurons with no hidden layer:
  - hand-written-characters of the figures from zero to nine drawn on the 10x10 pixels are to be recognized
  - by activating only the i-th neuron of the output remaining other neurons not activated.

□ From a biological analogy, the followings are defined.

Chromosome:
Each chromosome is comprised of 1000 *genes*, representing 1000 weights of one *individual* neural network.

Population:
A number of individuals, say 100, create a population.

Initialization:
All genes of all chromosomes are initialized at random, taking a value ranging from zero to one. This construct the first *generation*.

Fitness:
The ability of each individual is evaluated and assigned its fitness value. In our current problem, for example, each individual given a set of weights is evaluated by giving each output neuron 0.1 if the output is correct and 0 otherwise. Note that in the first generation, every individual performs rather badly but diferrently, since weights are determined at random.

Selection:
Two parents are selected to produce one offspring. Individuals of better performance should be more likely to be selected. For example, (1) selection is always from upper half of the population in which individuals have been sorted with an order of their fitness, or (2) selection is made sot that the probability is proportional to the fitness value. The production of an offspring is by employing the following two operations: *crossover* and *mutation*

Crossover:

A same position of the two parent chromosomes are determined at random (each time when they are selected), and cross them over.

Mutation:

For each gene of the offspring chromosome, produce one integer random number from 0 to 99. If the value is zero, then the gene is replaced with the other weight value also created at random each time.

□ Algorithm:

1. *Initialization:* Initialize all the weight values of all the individuals in the first generation at random so that it lies on $[0, 1]$.
2. *Evaluation:* Evaluate fitness of all the individuals of the population.
3. *Reproduction:*
   (1) Select two parents from the population.
   (2) Crossover the two parents and produce one offspring.
   (3) Mutate the offspring.
   (4) Repeat (1) to (3) above until new generation of same size is obtained.
4. Repeat 2. to 3. above until the perfect solution is obtained or prespecified generations is reached.

Chapter 2.

Features based on Linear Transformation.

- Generation of features using linear transforms of the input samples.
- Appropriate transform removes information redundancy and reduce feature space dimension, leading to information packing.

  e.g.,

  - Fourier coefficients as features:
    * most of the energy lies in the low-frequency components due to the high correlation between pixels and
    * we can neglect low-energy, high-frequency coefficient with little loss of information.

□ Fourier Coefficient (revisit).

We can expand any continuous function $f(x)$ as:

$$f(x) \simeq \sum_{k=1}^{\infty} a_k \cos kx + b_k sinkx \tag{12}$$

where we can calculate the Fourier coefficient a_k and b_k as

$$a_k = \int_{-\pi}^{\pi} f(x) \cos kx \, dx \tag{13}$$

$$b_k = \int_{-\pi}^{\pi} f(x) \sin kx \, dx \tag{14}$$

$$\tag{15}$$

(a_k and b_k are called higher frequency component for larger k and vice versa.)

- Exercise

  Obtain the Fourier expansion of the followings.

  (1) $y = x$

  (2) $y = x^2$

The other formulation

$$f(x) \simeq \sum_{k=-\infty}^{\infty} c_k \exp(-jkx) \tag{16}$$

where

$$c_k = \int_{-\pi}^{\pi} f(x) \exp(jkx) \, dx \tag{17}$$

- Exercise

  Verify this employing the relations

  $\exp(-jkx) = \cos kx + j \sin kx$

□ 1-D Discrete Fourier Transform.

- DFT:

$$y(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) \exp(-j\frac{2\pi}{N}kn), \quad k = 0, 1, \cdots, N-1. \tag{18}$$

- Inverse-DFT:

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y(k) \exp(j\frac{2\pi}{N}kn), \quad n = 0, 1, \cdots, N-1. \tag{19}$$

- An example of $N = 4$: DFT of $x(0), x(1), x(2), x(3)$
  e.g.

$$\begin{aligned} y(1) &= \frac{1}{2}\{x(0)\exp(0) + x(1)\exp(-j\frac{\pi}{2}) \\ &\quad +x(2)(-j\frac{2\pi}{2}) + x(3)\exp(-j\frac{3\pi}{2})\} \end{aligned} \tag{20}$$

$$\begin{pmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{pmatrix} \tag{21}$$

Or

$$\mathbf{y} = A\mathbf{x} \tag{22}$$

where

$$A_{mn} = \exp(-j\frac{2\pi}{N})(m-1)(n-1) \tag{23}$$

- Inverse Transformation

$$A^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \tag{24}$$

- Orthogonal linear transformation

In a linear transform

$$y(k) = \sum_{n=0}^{N-1} x(n)a_n(k), \tag{25}$$

if

$$a_m(k)a_n(k) = \delta_{mn} \tag{26}$$

holds, then the linear transformation is called *unitary* (*orthogonal* in real variables).

□ 2-D DFT

$$Y(k,l) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} X(m,n) \exp(-j\frac{2\pi}{N}km) \exp(-j\frac{2\pi}{N}ln) \tag{27}$$

$$X(m,n) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} Y(k,l) \exp(j\frac{2\pi}{N}km) \exp(j\frac{2\pi}{N}ln) \tag{28}$$

□ Other Linear Transformation

- HAAR
- HADAMAR
- WAVELET

(Though this won't be studied in this course.)

□ Hadamar Transform:

For a vector $\mathbf{x}$ of N samples where $N = 2^n$, the transform and its inverse transform are:

$$\mathbf{y} = H_n \mathbf{x}, \quad \mathbf{x} = H_n \mathbf{y} \tag{29}$$

where

$$H_n = H_1 \otimes H_{n-1} \tag{30}$$

and

$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{31}$$

Here $\otimes$ denotes the Kronecker product for two matrices:[1]

$$[A \otimes B]_{ij} = A_{ij} B \tag{35}$$

[1] for example, if

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \tag{32}$$

and

$$B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \tag{33}$$

then we define

$$A \otimes B = \begin{pmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{pmatrix} \tag{34}$$

□ Haar Transform:

Haar function $h_k(z)$, which is continuous and are defined in $[0,1]$, for the transform of $L = 2^n$ dimensional vector $\mathbf{x}$:

(1) Decomposition of the order k into two integers p and q.

$$k = 2^p + q - 1, \quad k = 0, 1, \cdots, L-1. \tag{36}$$

The decomposition is unique when

$$\begin{cases} q = 0 \quad \text{or} \quad 1 & \text{if} \quad p = 0 \\ 0 \leq p \leq n-1, \quad 0 < q \leq 2^p & \text{if} \quad p \neq 0 \end{cases} \tag{37}$$

(2) Haar functions are

$$h_0(z) \equiv h_{00}(z) = \frac{1}{\sqrt{L}} \tag{38}$$

and

$$h_k(z) \equiv h_{pq}(z) = \frac{1}{\sqrt{L}} \begin{cases} 2^{\frac{p}{2}} & \text{if } \frac{q-1}{2^p} \leq z < \frac{q-\frac{1}{2}}{2^p} \\ -2^{\frac{p}{2}} & \text{if } \frac{q-\frac{1}{2}}{2^p} \leq z < \frac{q}{2^p} \\ 0 & \text{otherwise in } [0,1] \end{cases} \tag{39}$$

(3) Then the Haar transform matrix of order L consists of rows of the Haar function computed at

$$z = \frac{m}{L}, \quad m = 0, 1, 2, \cdots L-1.$$

- Exercise

Obtain the 8×8 transform matrix.

- An example of $n = 3$, that is, $L = 8$.

$$k = 2^p + q - 1, \quad k = 0, 1, \cdots, 7 \tag{40}$$

where

$$\begin{cases} q = 0 \quad \text{or} \quad 1 & \text{if} \quad p = 0 \\ 0 \le p \le 2, \quad 0 < q \le 2^p & \text{if} \quad p \ne 0. \end{cases} \tag{41}$$

So,

p=0	q=0	k=0
	q=1	k=1
	$(0 < q \le 2)$	$(k = 1 + q)$
p=1	q=1	k=2
	q=2	k=3
	$(0 < q \le 4)$	$(k = 3 + q)$
p=2	q=1	k=4
	q=2	k=5
	q=3	k=6
	q=4	k=7

Hence,

k	0	1	2	3	4	5	6	7
p	0	0	1	1	2	2	2	2
k	0	1	1	2	1	2	3	4

Then the Haar functions are:

$$h_1(z) \equiv h_{01}(z) = \frac{1}{\sqrt{8}} \begin{cases} 1 & \text{if } -1 \le z < \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} \le z < 1 \end{cases} \tag{42}$$

$$h_2(z) \equiv h_{11}(z) = \frac{1}{\sqrt{8}} \begin{cases} \sqrt{2} & \text{if } 0 \le z < \frac{1}{4} \\ -\sqrt{2} & \text{if } \frac{1}{4} \le z < \frac{1}{2} \\ 0 & \text{otherwise in } [0,1] \end{cases} \tag{43}$$

$$h_3(z) \equiv h_{12}(z) = \frac{1}{\sqrt{8}} \begin{cases} \sqrt{2} & \text{if } \frac{1}{2} \le z < \frac{3}{4} \\ -\sqrt{2} & \text{if } \frac{3}{4} \le z < 1 \\ 0 & \text{otherwise in } [0,1] \end{cases} \tag{44}$$

$$\cdots\cdots$$

$$h_7(z) \equiv h_{24}(z) = \frac{1}{\sqrt{8}} \begin{cases} 2 & \text{if } \frac{3}{4} \le z < \frac{7}{8} \\ -2 & \text{if } \frac{7}{8} \le z < 1 \\ 0 & \text{otherwise in } [0,1] \end{cases} \tag{45}$$

and

$h_0(z)$	1	1	1	1	1	1	1	1
$h_1(z)$	1	1	1	1	–1	–1	–1	–1
$h_2(z)$	$\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$	0	0	0	0
$h_3(z)$	0	0	0	0	$\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$
$h_4(z)$	2	–2	0	0	0	0	0	0
$h_5(z)$	0	0	2	–2	0	0	0	0
$h_6(z)$	0	0	0	0	2	–2	0	0
$h_7(z)$	0	0	0	0	0	0	2	–2

Since

$$H_{ij} = \frac{1}{\sqrt{L}} h_j(\frac{j}{L}), \tag{46}$$

we obtain

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix} \tag{47}$$

□ Features for Texture Recognitions:

To describe an image by *fine* or *coarse*, *smooth* or *irregular*, *homogeneous* or *inhomogeneous* and so forth

- First-order Statistics Features
- Second-order Statistics Features

□ First-order Statistics Features:

Let

$$P(I) = \frac{\text{number of pixels with gray level } I}{\text{total number of pixels in the region}} \tag{48}$$

Assume that we have N_g gray levels.

- Then, the *moments*

$$m_i = E[I^i] = \sum_{I=0}^{N_g-1} I^i P(I) \tag{49}$$

- and the central moment

$$\mu_i = E[(I - E[I])^i] = \sum_{I=0}^{N_g-1} (I - m_I)^i P(I) \tag{50}$$

⋆ The most frequently used central moments:

· $\mu_2 = \sigma^2 \cdots$ *variance*

a measure of the histogram width:
how much the gray levels differ from the mean

· $\mu_3 \cdots$ *skewness*

the degree of histogram asymmetry around the mean

· $\mu_4 \cdots$ *kurtosis*

a measure of the histogram sharpness

- *platykurtic* (sharp);
- *mesokurtic* (medium);
- *leptokurtic* (dull).

- *Entropy*

$$H = -E[\log_2 P(I)] = -\sum_{I=0}^{N_g-1} P(I) \log_2 P(I) \qquad (51)$$

a measure of histogram uniformity:

The closer to the uniform distribution ($P(I) = \text{const.}$), the higher the H.

- Exercise

Calculate μ_1, μ_2, μ_3 and H of the four histograms bellow.

	histgram-1	histgram-2	histgram-3	histgram-4
$P(0)$	5	1	7	5
$P(1)$	6	14	8	5
$P(2)$	5	1	1	5
broad?	yes	no	medium	yes
symmetric?	yes	yes	no	yes
sharp?	no	yes	medium	no
uniform?	medium	no	no	yes

□ Second-order Statistics Features:

The first-order statistics provide information as for the gray level distribution of the image, but do not give any information about the relative positions of the gray levels within the image, such as

- Are all low-value gray levels positioned together?
- Are they interchanged with the high-value ones?

That is, the second-order statistics indicate how two specific level I_1 and I_2 occupy pixels in a line of slope ϕ separating with a distance d. Then, a two-dimensional histogram:

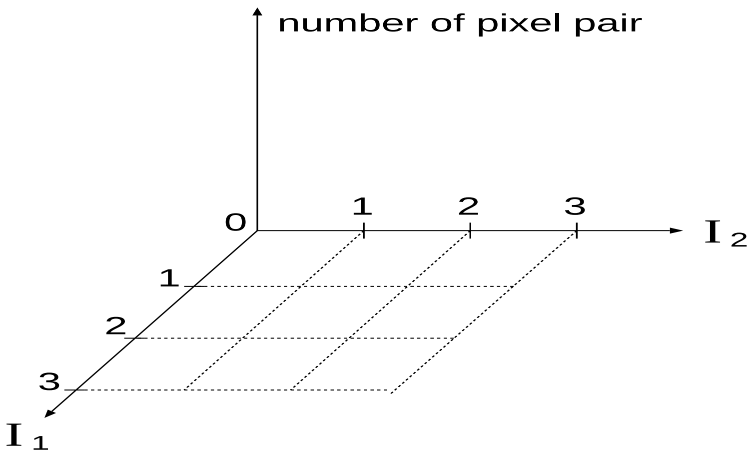
$$P(I_1, I_2) = \frac{\text{number of pairs of pixels with values } (I_1, I_2)}{\text{total number of possible pairs}} \tag{52}$$

with respect to

⋆ the relative distance d measured in pixel numbers

e.g., $d = 1$ for neighboring pixels.

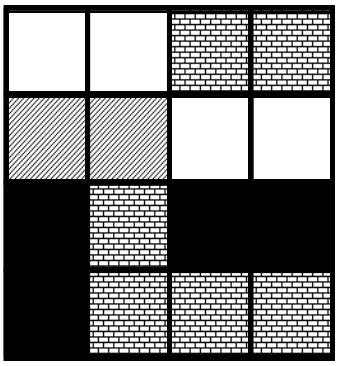
⋆ the orientation ϕ quantized in for directions: horizontal, diagonal, vertical, and anti-diagonal $(0°, 45°, 90°, 135°)$.



For each of the histograms the *co-occurrence matrix* is defined as:

$$A^{\phi}(d) = P(I_1, I_2) \tag{53}$$

Take for example the image with four gray levels bellow:



$$\begin{bmatrix} 0 & 0 & 2 & 2 \\ 1 & 1 & 0 & 0 \\ 3 & 2 & 3 & 3 \\ 3 & 2 & 2 & 2 \end{bmatrix}$$

	$d = 1$	$d = 1$	$d = 2$
	$\phi = 0°$	$\phi = 45°$	$\phi = 135°$
$P(0,0)$	2	0	2
$P(0,1)$	1	1	0
$P(0,2)$	1	2	0
$P(0,3)$	0	1	0
$P(1,1)$	1	0	2
$P(1,2)$	0	1	0
$P(1,3)$	0	1	2
$P(2,2)$	3	0	0
$P(2,3)$	3	3	0
$P(3,3)$	1	0	0
total sum	12	9	4

- The first one may be denoted as:

$$\frac{1}{12}\begin{pmatrix} 2 & 1 & 1 & 0 \\ \star & 1 & 0 & 0 \\ \star & \star & 3 & 3 \\ \star & \star & \star & 1 \end{pmatrix}. \tag{54}$$

However we usually adopt the way:

$$A^0(1) = \frac{1}{24}\begin{pmatrix} 4 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 6 & 3 \\ 0 & 0 & 3 & 2 \end{pmatrix}, \tag{55}$$

by making every i-j entry identical to j-i entry for $i \neq j$, and multiplying diagonal element by two for compensation.

□ Features for texture recognition

— 1. Features using *occurrence matrix.*

- Angular second moment:

$$X_{\text{asm}} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (P(i,j))^2 \tag{56}$$

⋆ A measure of the smoothness.

· If indeed all pixels are of the same gray level $I = k$, then $P(k,k) = 1$ and P(i,j)=0 for $i \neq k$ or $j \neq k$;

· Or, if we have all possible pairs of gray levels with equal probability $1/R$, then $X_{\text{asm}} = R/R^2 = 1/R$

⋆ The less smooth the region is, the more uniformly distributed $P(i,j)$ and the lower the X_{asm}.

- Contrast:

$$X_{\text{con}} = \sum_{n=0}^{N_g-1} n^2 \Big(\sum_{\substack{j=0 \\ |j-i|=n}}^{N_g-1} \sum_{i=0}^{N_g-1} P(i,j) \Big) \tag{57}$$

⋆ A measure of the image contrast.

· $\Sigma_i \Sigma_j P(i,j)$ is the percentage of pixel pairs whose intensity differs by n.

· The n^2 dependence weights the big differences more.

⋆ X_{con} takes high values for images of high contrast.

- Inverse difference moment:

$$X_{\text{idm}} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{P(i,j)}{1+(i-j)^2} \tag{58}$$

  ⋆ This feature takes high values for low-contrast images due to the inverse $(i-j)^2$ dependence.

- Entropy:

$$X_{\text{entropy}} = -\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i,j) \log_2 P(i,j) \tag{59}$$

  ⋆ A measure of randomness.

  ⋆ Higher values for more random images.

□ Run Length Features

— 2. Features using *gray level run lengths.*

- Gray Level Run:
  - a set of consecutive pixels having the same gray level.
- Run Length Features:
  - an information related to the number of each gray level appears in the image by itself, in pairs, and so on.
- Run Length Matrix Q^{ϕ}:
  - the (i-j) element is the number of times a gray level $(i-1)$ appears in the image with run length j.
  - the mesearment is along the line whose angle is ϕ from the horizontal line.

In the example we used before, i.e.

$$I = \begin{pmatrix} 0 & 0 & 2 & 2 \\ 1 & 1 & 0 & 0 \\ 3 & 2 & 3 & 3 \\ 3 & 2 & 2 & 2 \end{pmatrix} \tag{60}$$

$$Q^{0^\circ} = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{pmatrix} \quad \text{or} \quad Q^{45^\circ} = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{pmatrix} \tag{61}$$

□ Five Run Length Features:

Under the assumption that we have N_g gray levels and the largest possible run length in the image is N_r, we define them as follows:

- Short run emphasis:

$$Y_{\text{sre}} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} (Q(i,j)/j^2)}{\text{total number of run length}} \tag{62}$$

- Long run emphasis:

$$Y_{\text{lre}} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} (Q(i,j)j^2)}{\text{total number of run length}} \tag{63}$$

- Gray level non-uniformity:

$$Y_{\text{gln}} = \frac{\sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_r} Q(i,j)\right)^2}{\text{total number of run length}} \tag{64}$$

- Run length non-uniformity:

$$Y_{\text{gln}} = \frac{\sum_{j=1}^{N_r} \left(\sum_{i=1}^{N_g} Q(i,j)\right)^2}{\text{total number of run length}} \tag{65}$$

- Run percentage:

$$Y_{\text{rp}} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} Q(i,j)}{\text{total possible runs in the image}} \tag{66}$$

□ Some remarks:

- The total number of run lengths in the matrix is

$$\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} Q(i,j) \tag{67}$$

  which takes the value 9 for $\phi = 0°$, and 16 for $\phi = 45°$ in the equation (61).

- Short run emphasis:
  - Y_{sre} emphasizes small run lengths due to the division by j^2;
  - Thus Y_{sre} is expect to be large for coarser image.
- Long run emphasis:
  - Y_{lre} gives emphasis to long run lengths due to the division by j^2;
  - Thus Y_{lre} is expect to be large for smoother image.
- Gray level non-uniformity:
  - The term $\Sigma_j^{N_r} Q(i,j)$ is the total number of run length for each gray level.
  - Large run length values contribute a great deal becaus of the square.
  - When runs are uniformly distributed among the gray levels, Y_{gln} takes small values.
- Run length non-uniformity:
  - In a similar way as Y_{gln}, Y_{rln} is a measure of run length nonuniformity.

- Run percentage:
  - If all runs had length equal to one, that is, the total number of pixels Y_{rp} takes low values for smooth images.

□ Examples:

For an image:

$$I = \begin{pmatrix} 0 & 0 & 2 & 2 \\ 1 & 1 & 0 & 0 \\ 3 & 2 & 3 & 3 \\ 3 & 2 & 2 & 2 \end{pmatrix} \tag{68}$$

we have[2]

$$Q^0 = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{pmatrix} \tag{69}$$

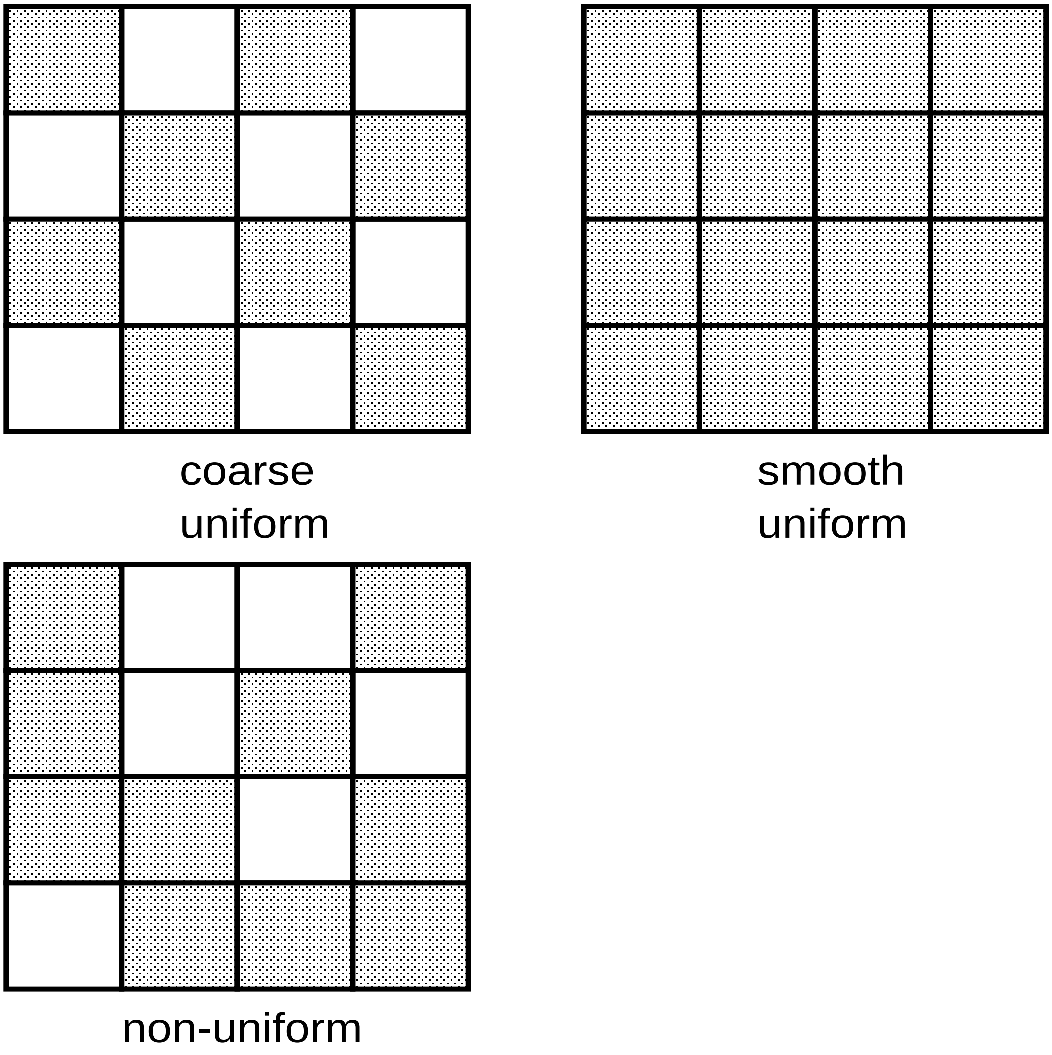
and

$$\begin{aligned} Y_{\text{sre}} &= \frac{2/4 + 1/4 + \{1 + 1/4 + 1/9\} + \{2 + 1/4\}}{9} \\ Y_{\text{lre}} &= \frac{\{2 \cdot 4\} + \{1 \cdot 4\} + \{1 + 1 \cdot 4 + 1 \cdot 9\}\{2 + 1 \cdot 4\}}{9} \\ Y_{\text{gln}} &= \frac{4 + 1 + 9 + 9}{9} \\ Y_{\text{rln}} &= \frac{9 + 25 + 1 + 0}{9} \\ Y_{\text{rp}} &= \frac{9}{16} \end{aligned} \tag{70}$$

[2] Here we calculate with Q^0, but usually we use $\{Q^0 + Q^{45} + Q^{90} + Q^{135}\}/4$. Guess the reason.

☐ Coarse, smooth, uniform?:

An example of $N_g = 2$:



□ Bayes Decision Theory

— for the design of classifiers.

As we have learned so far,

classifiers classify unknown patterns into the *most probable class.*

Then

what "most probable" means?

- Moreover

  Pattern Recognition is more or less statistical due to:

  ∗ statistic nature of feature selection;

  ∗ statistic variation of patterns;

  ∗ noise in employing sensors.

Hence

we study here the Baysean Statistical Theory.

- When we classify an unknown pattern which is represented by a feature vector $\mathbf{x}$, we consider M conditional probability

$$p(\omega_i|\mathbf{x}) \quad i = 1, \cdots, M \tag{71}$$

called *a posteriori probability*.

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

- The other statistical quantities:
  - *a priori* probability $p(\omega_i)$;
  - *likelihood* function of ω_i w.r.t. $\mathbf{x}$;
  - probability distribution function (pdf) of $\mathbf{x}$.

- Baysian rule is

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{\sum_{k=1}^{M} p(\mathbf{x}|\omega_k)p(\omega_k)} \tag{72}$$

where the denominator

$$\sum_{k=1}^{M} p(\mathbf{x}|\omega_k)p(\omega_k) = p(\mathbf{x}). \tag{73}$$

- Example-1:
  - We have two bags which we can not tell the difference from outside.
  - One bag called R includes 70 red balls and 30 blue balls.
  - The other bag called B includes 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 is the number of red balls was 8 and blue one was 4.
  - 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

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

(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 two out of 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 aboutme. So please tell that."
  - The guard answers **C**. $\Rightarrow$ data D
  - Now, **A** knows one of **A** or **B** is sure to be free.

So you guess *a posteriori* probability $p(A|D) = 1/2$?

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

Try to apply Baysean rule, if you agree

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

$$p(D|A) = 1/2 \tag{74}$$
$$p(D|B) = 1 \tag{75}$$
$$p(D|C) = 0 \tag{76}$$

Theorem 1 (Bayes Decision Theory) *A pattern which has* $\mathbf{x}$ *should be classified so that* $p(\omega_i|\mathbf{x})$ *takes the maximum value, where*

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{\sum_{k=1}^{M} p(\mathbf{x}|\omega_k)p(\omega_k)} \tag{77}$$

- $p(\omega_i)$ are usually, known or if not it can be estimate from the training sample, i.e.,
  - if N_i patterns out of N test samples are classified to ω_i then
  $$p(\omega_i) = \frac{N_i}{N} \tag{78}$$
- $p(\mathbf{x}|\omega_i)$ can be also estimated from the training samples (later in detail.)

□ To summarize

- Bayes classifier is:

  If $p(\omega_i|\mathbf{x})$ is greater than any $p(\omega_j|\mathbf{x}) \quad j = 1, \cdots, M j \neq i$, then classify $\mathbf{x}$ in class ω_i

- If ω_i is adjacent to ω_j

  $p(\omega_i|\mathbf{x}) - p(\omega_j|\mathbf{x}) = 0$ is called a *decision surface*

- or

  if $f(\cdot)$ exists such that

$$g_i(x) \equiv f(p(\omega_j|\mathbf{x}) \tag{79}$$

  where $f(\cdot)$ is a monotonically increasing function, then f is called a discriminant function.

- Then

  if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for $\forall j \neq i$ then classify $\mathbf{x}$ to ω_i,

- and

  decision surface is determined by

$$g_{ij}(\mathbf{x}) \equiv g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0, \quad i, j = 1, 2, \cdots, M. \tag{80}$$

- But usually

  we know or can obtain $p(\mathbf{x}|\omega_j)$ rather than $p(\omega_j|\mathbf{x})$

- and

  in most cases $p(\mathbf{x}|\omega_j)$ is Gaussian distribution.

- Therefore

  if we use the equation (77) under an assamption of $p(\omega_1) = p(\omega_2) = \cdots = p(\omega_N) = 1/N$ we can use $p(\mathbf{x}|\omega_j)$.

□ An example for $i = 2$.

$$\begin{cases} \text{if } p(\omega_1|\mathbf{x}) > p(\omega_2|\mathbf{x}) & \mathbf{x} \text{ is classified to } \omega_1. \\ \text{if } p(\omega_2|\mathbf{x}) > p(\omega_1|\mathbf{x}) & \mathbf{x} \text{ is classified to } \omega_2. \end{cases} \tag{81}$$

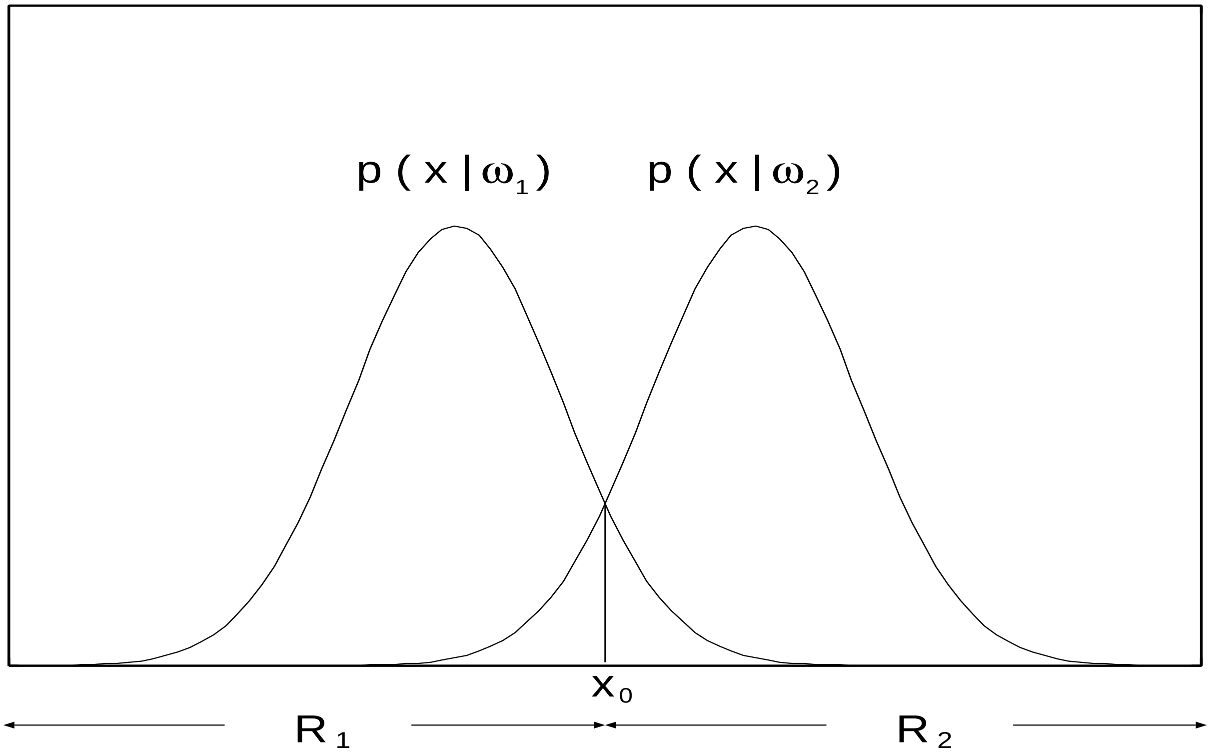
where

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{p(\mathbf{x}|\omega_1)p(\omega_1) + p(\mathbf{x}|\omega_2)p(\omega_2)}$$

so, if $p(\omega_1) = p(\omega_2) = 1/2$

$$\begin{cases} \text{if } p(\mathbf{x}|\omega_1) > p(\mathbf{x}|\omega_2) & \mathbf{x} \text{ is classified to } \omega_1. \\ \text{if } p(\mathbf{x}|\omega_2) > p(\mathbf{x}|\omega_1) & \mathbf{x} \text{ is classified to } \omega_2. \end{cases} \tag{82}$$

where $p(\mathbf{x}|\omega_i)$ is a probability density function.



Classification Error Probability in this case

$$p_e = \int_{-\infty}^{x_0} p(\mathbf{x}|\omega_2)dx + \int_{x_0}^{\infty} p(\mathbf{x}|\omega_1)dx \tag{83}$$

- The equation (83) is also expressed as:

$$p_e = \int_{R_1} p(\mathbf{x}|\omega_2)dx + \int_{R_2} p(\mathbf{x}|\omega_1)dx \tag{84}$$

where $R_i, \quad i = 1, 2$ are regions shown in the Figure.

- By generalizing this, we obtain the probability of misclassify ω_k to ω_i as:

$$p_k = \sum_{i=1}^{M} \int_{R_i} p(\mathbf{x}|\omega_k)dx \tag{85}$$

- If λ_{ki}, the importance of misclassification of ω_k to ω_i, is taken into consideration, then the *total risk* are defined as:

$$p_k = \sum_{i=1}^{M} \lambda_{ki} \int_{R_i} p(\mathbf{x}|\omega_k)dx \tag{86}$$

- And $p = \sum_{k=1}^{M} r_k p(\omega_k)$ should be minimized.

- This is achieved when each integral is minimized.

- So, $\mathbf{x}$ should be classified to ω_i, that is, $\mathbf{x} \in R_i \quad$ if

$$\sum_{k=1}^{M} \lambda_{ki} p(\mathbf{x}|\omega_k)p(\omega_k) < \sum_{k=1}^{M} \lambda_{kj} p(\mathbf{x}|\omega_k)p(\omega_k) \tag{87}$$

for $\forall j \neq i$.

- In a two-class case example:
  - assign $\mathbf{x}$ to ω_1 if

$$(\lambda_{21} - \lambda_{22})p(\mathbf{x}|\omega_2)p(\omega_2) < (\lambda_{12} - \lambda_{11})p(\mathbf{x}|\omega_1)p(\omega_1) \tag{88}$$

  - Therefore, assign $\mathbf{x}$ to ω_1 if

$$\frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x}|\omega_2)} > \frac{p(\omega_1)}{p(\omega_2)}\frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}} \tag{89}$$

  - In the special case where

$$\lambda_{11} = \lambda_{22} = 0, \quad \lambda_{21} > \lambda_{12}, \quad \text{and} p(\omega_1) = p(\omega_2) = 1/2$$

  assign $\mathbf{x}$ to ω_2 if

$$p(\mathbf{x}|\omega_2) > p(\mathbf{x}|\omega_1)\frac{\lambda_{12}}{\lambda_{21}} \tag{90}$$

  - What is the implication of equation (90)?

- Example

  ⋆ $p(\mathbf{x}|\omega_1)$ and $p(\mathbf{x}|\omega_1)$ follows a Gaussian distribution whose mean and standard deviation are:

$$\mu_1 = 0, \quad \mu_2 = 1, \quad \text{and} \sigma_1 = \sigma_2 = 1/2.$$

  that is,

$$\begin{aligned} p(\mathbf{x}|\omega_1) &= \frac{1}{\sqrt{\pi}} \exp(-x^2) \\ p(\mathbf{x}|\omega_1) &= \frac{1}{\sqrt{\pi}} \exp(-x^2) \end{aligned}$$

(1) Obtain x_0.

(2) When

$$[\lambda_{ik}] = \begin{pmatrix} 0 & 0.5 \\ 1.0 & 0 \end{pmatrix}$$

how x_0 will be affected?

□ Multi-dimensional Gaussian distribution

Classifications in real-world problems, however, might be in a multi-dimensional space, not 1-D space.

So, we must be familiar with multi-dimensional Gaussian distribution.

The one-dimensional version

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{91}$$

is expanded for $\mathbf{x} = (x_1, x_2, \cdots, x_l)$ to the form:

$$p(x) = \frac{1}{\sqrt{2\pi}|S|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T S^{-1}(\mathbf{x}-\mathbf{m})) \tag{92}$$

where the components of $\mathbf{m}$ are the mean of $x_i \quad i = 1, \cdots, l)$, that is,

$$m_i = E[x_i] \tag{93}$$

and S is called a *covariance matrix* and the components are

$$\begin{cases} \sigma_{ii} = \sigma_i^2 = E[(x_i - m_i)^2] \\ \sigma_{ij} = E[(x_i - m_i)(x_j - m_j)] \quad \text{if} \quad i \neq j \end{cases} \tag{94}$$

- If we use here ln as $f(\cdot)$ in the equation (79)

$$\begin{aligned} g_i(\mathbf{x}) &= \ln(p(\mathbf{x}|\omega_i)p(\omega_i)) \\ &= \ln p(\mathbf{x}|\omega_i) + \ln p(\omega_i) \\ &= -\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T S_i^{-1}(\mathbf{x}-\mathbf{m}_i) + \ln p(\omega_i) + c_i \\ &= -\frac{1}{2}\mathbf{x}^T S_i^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^T S_i^{-1}\mathbf{m}_i - \frac{1}{2}\mathbf{m}_i^T S_i^{-1}\mathbf{m}_i \\ &\qquad + \frac{1}{2}\mathbf{m}_i^T S_i^{-1}\mathbf{x} + \ln p(\omega_i) + c_i \end{aligned} \tag{95}$$

where

$$c_i = -\frac{l}{2}\ln 2\pi - \frac{1}{2}\ln|S_i| \tag{96}$$

- If for instance

$$l = 2, \quad \text{and} \quad S_i = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \tag{97}$$

then

$$\begin{aligned} g_i(\mathbf{x}) &= -\frac{1}{2\sigma_i^2}(x_1^2 + x_2^2) + \frac{1}{\sigma_i^2}(\mu_{i1}x_1 + \mu_{i2}x_2) \\ &\qquad - \frac{1}{2\sigma_i^2}(\mu_{i1}^2 + \mu_{i2}x^2) + \ln p(\omega_i) + c_i \end{aligned} \tag{98}$$

- This is a *quadric* form of x_i and x_2, that is, the classification space is either of
  - ⋆ *ellipsoids*
  - ⋆ *parabola*
  - ⋆ *hyperbola*
  - ⋆ *pair of lines*

- Two examples:

$$1)\quad S_1 = \begin{pmatrix} 0.10 & 0 \\ 0 & 0.15 \end{pmatrix} \qquad S_2 = \begin{pmatrix} 0.20 & 0 \\ 0 & 0.25 \end{pmatrix}$$

$$2)\quad S_1 = \begin{pmatrix} 0.10 & 0 \\ 0 & 0.15 \end{pmatrix} \qquad S_2 = \begin{pmatrix} 0.15 & 0 \\ 0 & 0.10 \end{pmatrix}$$