Digital Image Processing

Object Recognition

Christophoros Nikou

cnikou@cs.uoi.gr

Images taken from: R. Gonzalez and R. Woods. Digital Image Processing, Prentice Hall, 2008

Object Recognition

One of the most interesting aspects of the world is that it can be considered to be made up of patterns.

A pattern is essentially an arrangement.
It is characterized by the order of the elements of which it is made, rather than by the intrinsic nature of these elements

Norbert Wiener

Introduction

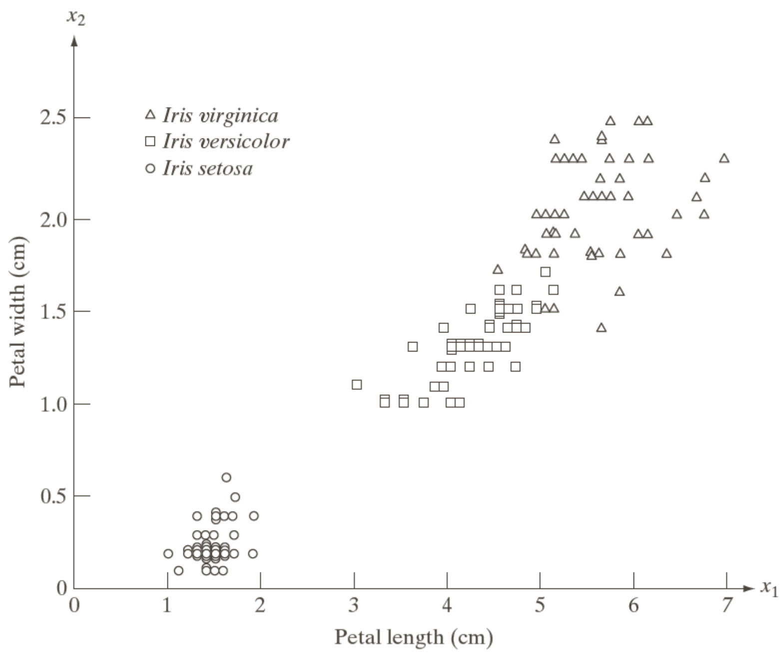
- Recognition of individual image regions (*objects* or *patterns*).
- Introduction to basic techniques.
  - Decision theoretic techniques.
    - Quantitative descriptors (e.g. area, length…).
    - Patterns arranged in vectors.
  - Structural techniques.
    - Qualitative descriptors (relational descriptors for repetitive structures, e.g. staircase).
    - Patterns arranged in strings or trees.
- Central idea: **Learning** from sample patterns.

Patterns and pattern classes

- Pattern: an arrangement of *descriptors* (or *features*).
- Pattern class: a family of patterns sharing some common properties.
  - They are denoted by $\omega_1, \omega_2, \ldots, \omega_W$, W being the number of classes.
- Goal of pattern recognition: assign patterns to their classes with as little human interaction as possible.

Pattern vectors

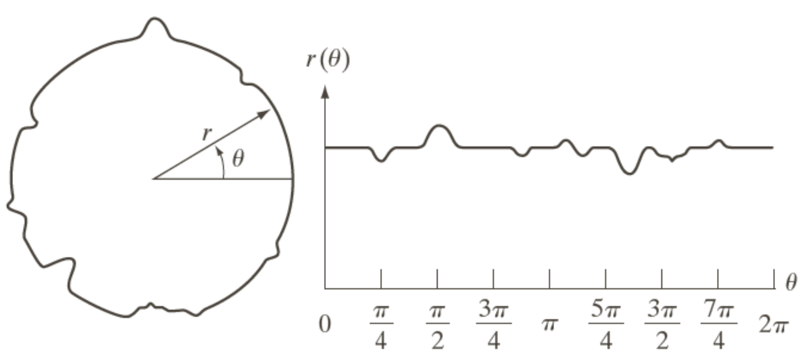
- Historical example
  - Recognition of three types of iris flowers by the lengths and widths of their petals (Fisher 1936).
- Variations between and within classes.
- Class separability depends strongly on the choice of descriptors.



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Pattern vectors (cont.)

- Shape signature represented by the sampled amplitude values.
- Cloud of n-dimensional points.
- Other shape characteristics could have been employed (e.g. moments).
- The choice of descriptors has a profound role in the recognition performance.



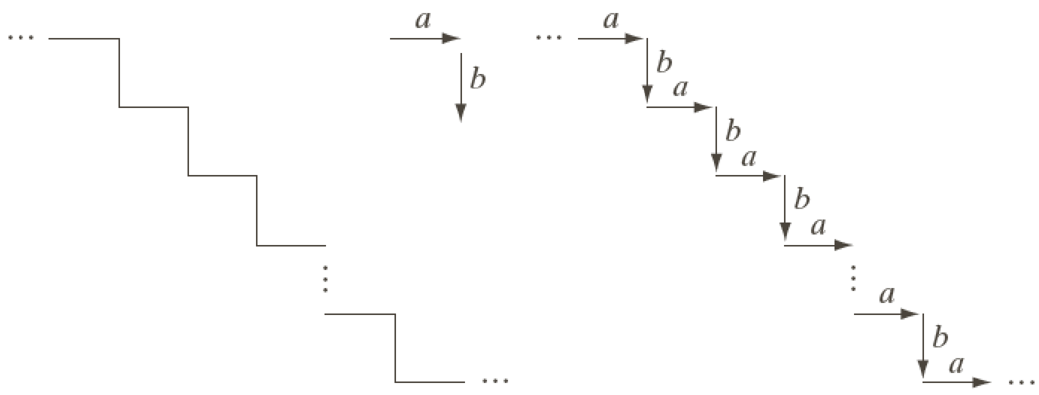
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

String descriptors

- Description of structural relationships.
- Example: fingerprint recognition.
- Primitive components that describe. fingerprint ridge properties.
  - Interrelationships of print features (*minutiae*).
    - Abrupt endings, branching, merging, disconnected segments,…
  - Relative sizes and locations of print features.

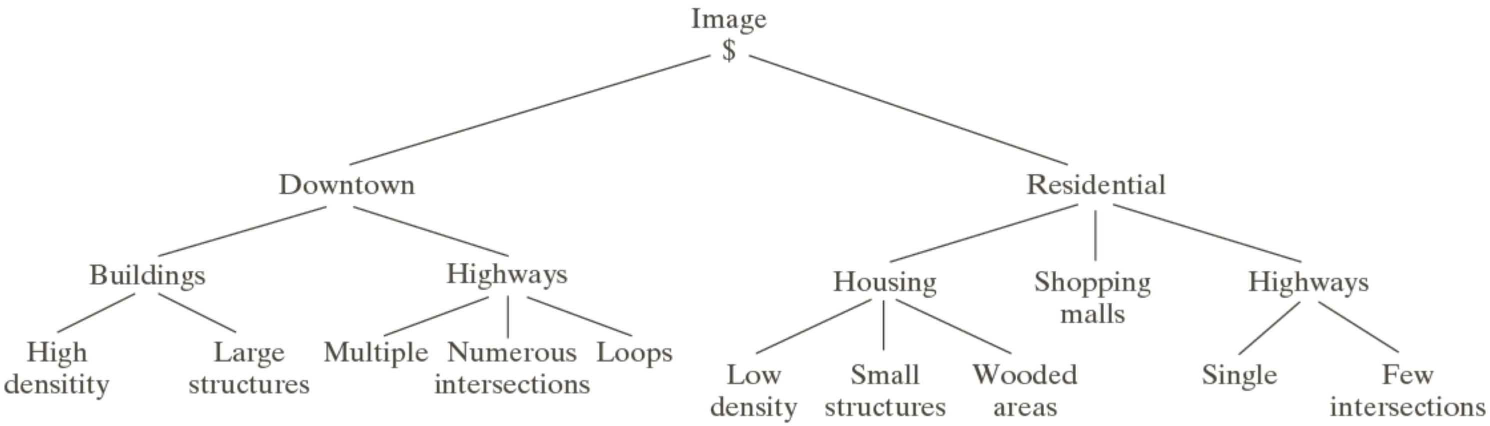
String descriptors (cont.)

- Staircase pattern described by a head-to-tail structural relationship.
- The rule allows only alternating pattern.
- It excludes other types of structures.
- Other rules may be defined.



Tree descriptors

- Hierarchical ordering
- In the satelite image example, the structural relationship is defined as: “composed of”



Decision-theoretic methods

- They are based on *decision* (*discriminant*) functions.
- Let $\mathbf{x}=[x_1, x_2,\ldots, x_n]^T$ represent a pattern vector.
- For W pattern classes $\omega_1, \omega_2,\ldots, \omega_W$, the basic problem is to find W decision functions

$$d_1(\mathbf{x}), d_2(\mathbf{x}),\ldots, d_W(\mathbf{x})$$

with the property that if $\mathbf{x}$ belongs to class ω_i:

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \text{ for } j = 1, 2, \ldots, W; \quad j \neq i$$

Decision-theoretic methods (cont.)

- The decision boundary separating class ω_i from class ω_j is given by the values of $\mathbf{x}$ for which $d_i(\mathbf{x}) = d_j(\mathbf{x})$ or

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

- If $\mathbf{x}$ belongs to class ω_i:

$$d_{ij}(\mathbf{x}) > 0 \quad \text{for} \quad j = 1, 2, \ldots, W; \quad j \neq i$$

Decision-theoretic methods (cont.)

- Matching: an unknown pattern is assigned to the class to which it is closest with respect to a metric.
  - Minimum distance classifier
    - Computes the Euclidean distance between the unknown pattern and each of the prototype vectors.
  - Correlation
    - It can be directly formulated in terms of images
- Optimum statistical classifiers
- Neural networks

Minimum distance classifier

The prototype of each pattern class is the mean vector:

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}_j \quad j = 1, 2, ..., W$$

Using the Euclidean distance as a measure of closeness:

$$D_j(\mathbf{x}) = \left\| \mathbf{x} - \mathbf{m}_j \right\| \quad j = 1, 2, ..., W$$

We assign $\mathbf{x}$ to class ω_j if $D_j(\mathbf{x})$ is the smallest distance. That is, the smallest distance implies the best match in this formulation.

Minimum distance classifier (cont.)

It is easy to show that selecting the smallest distance is equivalent to evaluating the functions:

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \quad j = 1, 2, \ldots, W$$

and assigning $\mathbf{x}$ to class ω_j if $d_j(\mathbf{x})$ yields the **largest** numerical value. This formulation agrees with the concept of a decision function.

Minimum distance classifier (cont.)

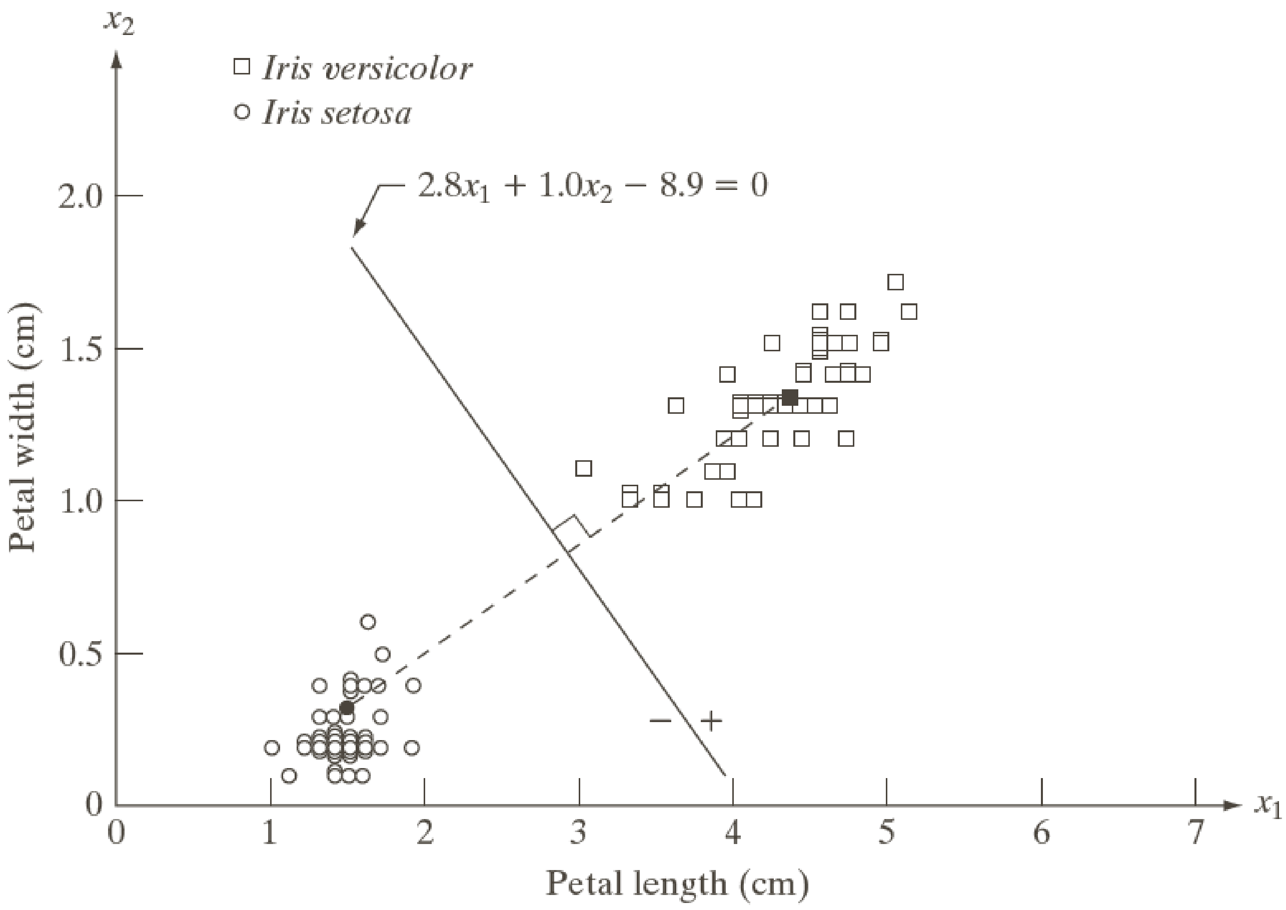
- The decision boundary between classes ω_i and ω_j is given by:

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x})$$

$$= \mathbf{x}^T(\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2}(\mathbf{m}_i - \mathbf{m}_j)^T(\mathbf{m}_i - \mathbf{m}_j) = 0$$

- The surface is the perpendicular bisector of the line segment joining $\mathbf{m}_i$ and $\mathbf{m}_j$.
- For $n=2$, the perpendicular bisector is a line, for $n=3$ it is a plane and for $n>3$ it is called a *hyperplane*.

Minimum distance classifier (cont.)

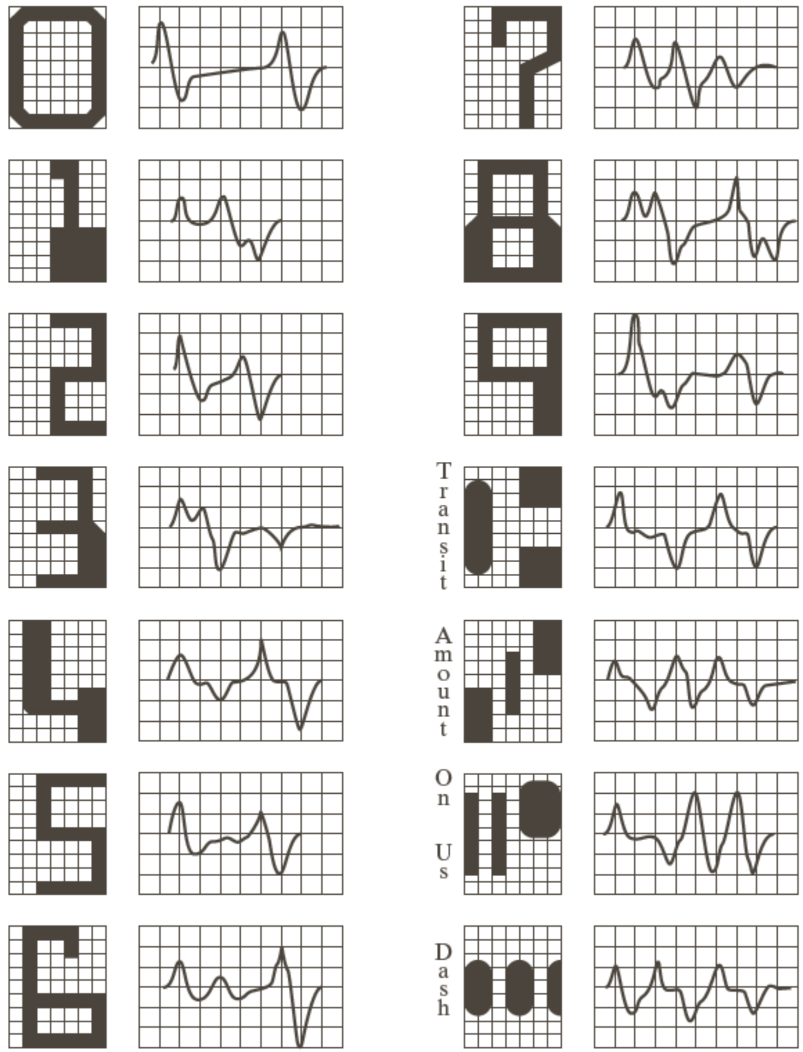


Minimum distance classifier (cont.)

- In practice, the classifier works well when the distance between means is large compared to the spread of each class.
- This occurs seldom unless the system designer controls the nature of the input.
- An example is the recognition of characters on bank checks
  - American Banker's Association E-13B font character set.

Minimum distance classifier (cont.)

- Characters designed on a 9x7 grid.
- The characters are scanned horizontally by a head that is narrower but taller than the character which produces a 1D signal proportional to the rate of change of the quantity of the ink.
- The waveforms (signatures) are different for each character.



Matching by correlation

- We have seen the definition of correlation and its properties in the Fourier domain.

$$g(x,y)=\sum_{s}\sum_{t}w(s,t)f(x+s,y+t)$$

$$\Leftrightarrow G(u,v)=F^{*}(u,v)W(u,v)$$

- This definition is sensitive to scale changes in both images.
- Instead, we use the *normalized correlation coefficient.*

Matching by correlation (cont.)

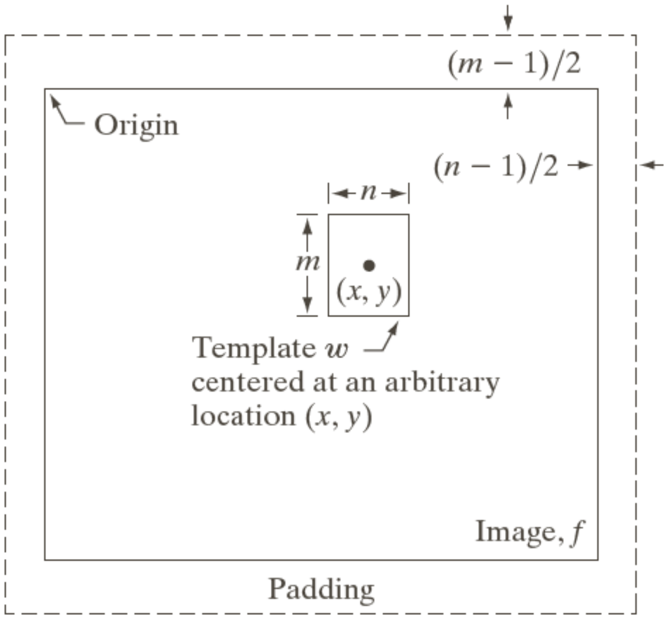
- Normalized correlation coefficient:

$$\gamma(x,y)=\frac{\sum_s\sum_t\left[w(s,t)-\bar{w}\right]\left[f(x+s,y+t)-\bar{f}_{xy}\right]}{\left\{\sum_s\sum_t\left[w(s,t)-\bar{w}\right]^2\sum_s\sum_t\left[f(x+s,y+t)-\bar{f}_{xy}\right]^2\right\}^{\frac{1}{2}}}$$

- $\gamma(x,y)$ takes values in $[-1,1]$.
- The maximum occurs when the two regions are identical.

Matching by correlation (cont.)

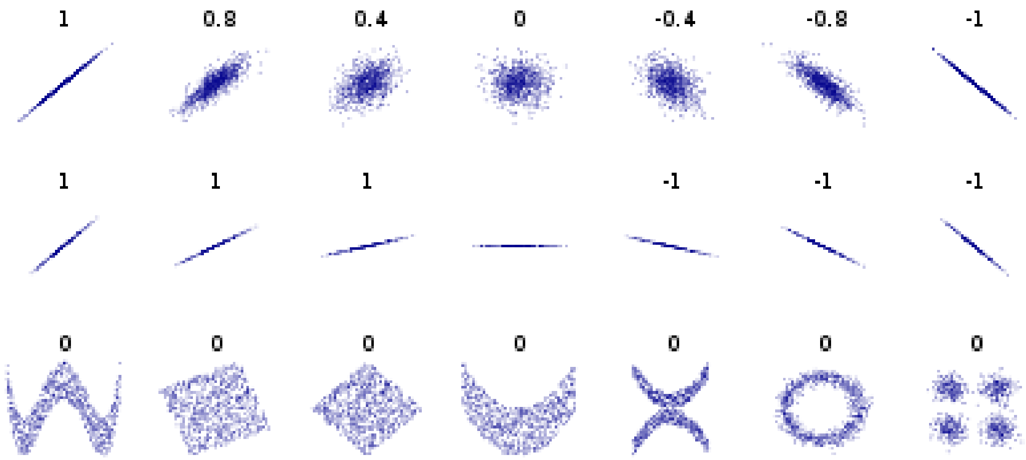
$$\gamma(x,y)=\frac{\sum_s\sum_t[w(s,t)-\bar{w}]\left[f(x+s,y+t)-\bar{f}_{xy}\right]}{\left\{\sum_s\sum_t[w(s,t)-\bar{w}]^2\sum_s\sum_t\left[f(x+s,y+t)-\bar{f}_{xy}\right]^2\right\}^{\frac{1}{2}}}$$



- It is robust to changes in the amplitudes.
- Normalization with respect to scale and rotation is a challenging task.

Matching by correlation (cont.)

- The compared windows may be seen as random variables.
- The correlation coefficient measures the linear dependence between X and Y.



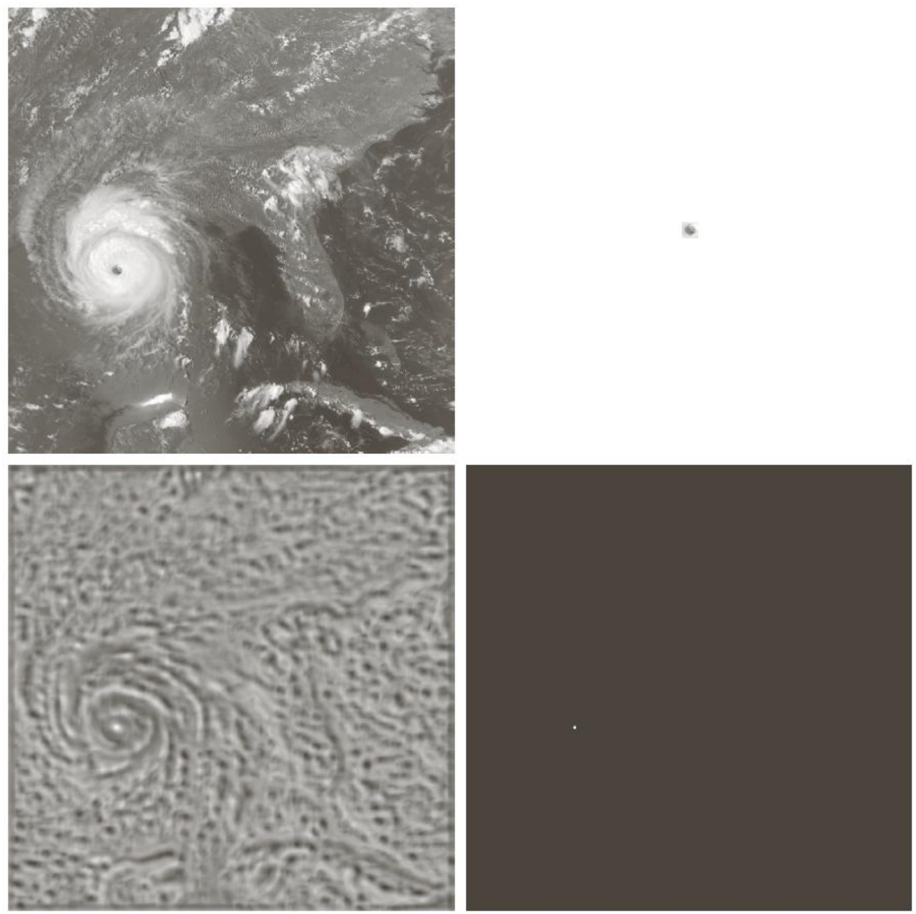
$$\gamma(X,Y) = \frac{E[(X - m_X)(Y - m_Y)]}{\sigma_X \sigma_X}$$

Matching by correlation (cont.)

Detection of the eye of the hurricane

Image

Template



Correlation coefficients

Best match

Optimum statistical classifiers

- A probabilistic approach to recognition.
- It is possible to derive an optimal approach, in the sense that, on average, it yields the lowest probability of committing classification errors.
- The probability that a pattern $\mathbf{x}$ comes from class ω_j is denoted by $p(\omega_j/\mathbf{x})$.
- If the classifier decides that $\mathbf{x}$ came from ω_j when it actually came from ω_i it incurs a loss denoted by L_{ij}.

Bayes classifier (cont.)

- As pattern $\mathbf{x}$ may belong to any of W classes, the *average loss* assigning $\mathbf{x}$ to ω_j is:

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj}\, p(\omega_k / \mathbf{x}) = \frac{1}{p(\mathbf{x})} \sum_{k=1}^{W} L_{kj}\, p(\mathbf{x} / \omega_k) P(\omega_k)$$

Because $1/p(\mathbf{x})$ is positive and common to all $r_j(\mathbf{x})$ the expression reduces to:

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj}\, p(\mathbf{x} / \omega_k) P(\omega_k)$$

Bayes classifier (cont.)

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj}\, p(\mathbf{x}/\omega_k) P(\omega_k)$$

- $p(\mathbf{x}/\omega_j)$ is the pdf of patterns of class ω_j (class conditional density).
- $P(\omega_j)$ is the probability of occurrence of class ω_j (*a priori* or prior probability).
- The classifier evaluates $r_1(\mathbf{x})$, $r_2(\mathbf{x})$,…, $r_W(\mathbf{x})$, and assigns pattern $\mathbf{x}$ to the class with the smallest average loss.

Bayes classifier (cont.)

- The classifier that minimizes the total average loss is called the *Bayes classifier.*
- It assigns an unknown pattern $\mathbf{x}$ to classs ω_i if:

$$r_i(\mathbf{x}) < r_j(\mathbf{x}), \text{ for } j = 1, 2, \ldots, W$$

or

$$\sum_{k=1}^{W} L_{ki}\, p(\mathbf{x}/\omega_k) P(\omega_k) < \sum_{q=1}^{W} L_{qj}\, p(\mathbf{x}/\omega_q) P(\omega_q), \text{ for all } j,\ j \neq i$$

Bayes classifier (cont.)

- The loss for a wrong decision is generally assigned to a non zero value (e.g. 1)
- The loss for a correct decision is 0.

$$L_{ij} = 1 - \delta_{ij}$$

Therefore,

$$r_j(\mathbf{x}) = \sum_{k=1}^{W} L_{kj}\, p(\mathbf{x} / \omega_k) P(\omega_k) = \sum_{k=1}^{W} (1 - \delta_{jk})\, p(\mathbf{x} / \omega_k) P(\omega_k)$$

$$= p(\mathbf{x}) - p(\mathbf{x} / \omega_j) P(\omega_j)$$

Bayes classifier (cont.)

- The Bayes classifier assigns pattern **x** to classs ω_i if:

$$p(\mathbf{x}) - p(\mathbf{x}/\omega_i)P(\omega_i) < p(\mathbf{x}) - p(\mathbf{x}/\omega_j)P(\omega_j)$$

or

$$p(\mathbf{x}/\omega_i)P(\omega_i) > p(\mathbf{x}/\omega_j)P(\omega_j)$$

which is the computation of decision functions:

$$d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j), \quad j = 1, 2, ..., W$$

It assigns pattern **x** to the class whose decision function yields the largest numerical value.

Bayes classifier (cont.)

- The probability of occurrence of each class $P(\omega_j)$ must be known.
  - Generally, we consider them equal, $P(\omega_j)=1/W$.

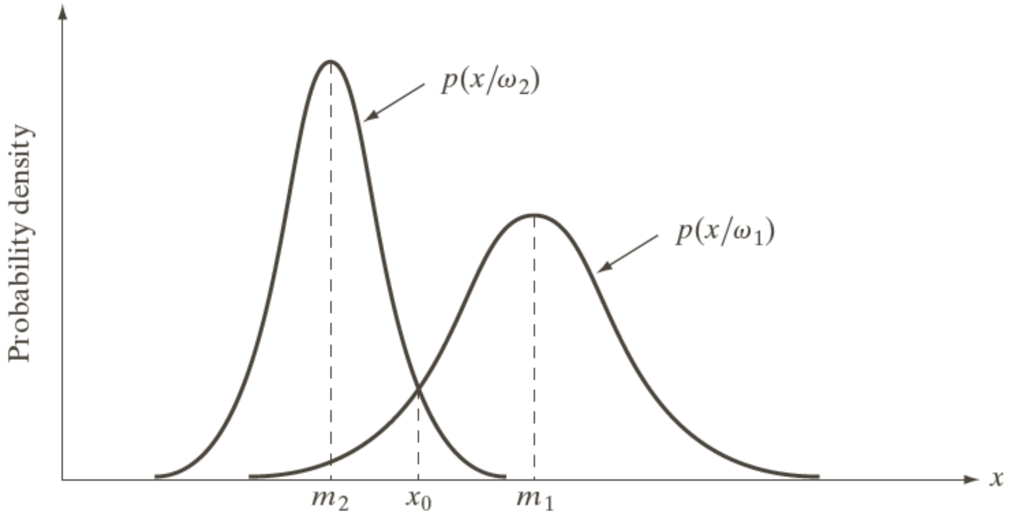
- The probability densities of the patterns in each class $P(\mathbf{x}/\omega_j)$ must be known.
  - More difficult problem (especially for multidimensional variables) which requires methods from pdf estimation.
  - Generally, we assume:
    - Analytic expressions for the pdf.
    - The pdf parameters may be estimated from sample patterns.
    - The Gaussian is the most common pdf.

Bayes classifier for Gaussian pattern classes

- We first consider the 1-D case for W=2 classes.

$$d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(\omega_j), \quad j = 1, 2, ..., W$$

- For $P(\omega_j)$=1/2:



Bayes classifier for Gaussian pattern classes (cont.)

- In the n-D case:

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}\left|\mathbf{C}_j\right|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}$$

- Each density is specified by its mean vector and its covariance matrix:

$$\mathbf{m}_j = E_j[\mathbf{x}]$$

$$\mathbf{C}_j = E_j[(\mathbf{x}-\mathbf{m}_j)(\mathbf{x}-\mathbf{m}_j)^T]$$

Bayes classifier for Gaussian pattern classes (cont.)

- Approximation of the mean vector and covariance matrix from samples from the classes:

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}$$

$$\mathbf{C}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} (\mathbf{x} - \mathbf{m}_j)(\mathbf{x} - \mathbf{m}_j)^T$$

Bayes classifier for Gaussian pattern classes (cont.)

- It is more convenient to work with the natural logarithm of the decision function as it is monotonically increasing and it does not change the order of the decision functions:

$$d_j(\mathbf{x}) = \ln\Big(p(\mathbf{x}/\omega_j)P(\omega_j)\Big) = \ln p(\mathbf{x}/\omega_j) + \ln P(\omega_j)$$

$$= \ln P(\omega_j) - \frac{1}{2}\ln\left|\mathbf{C}_j\right| - \frac{1}{2}(\mathbf{x}-\mathbf{m}_j)\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)^T$$

- The decision functions are hyperquadrics.

Bayes classifier for Gaussian pattern classes (cont.)

- If all the classes have the same covarinace $\mathbf{C}_j=\mathbf{C}$, $j=1,2,\ldots,W$ the decision functions are linear (*hyperplanes*):

$$d_j(\mathbf{x}) = \ln P(\omega_j) + \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{C}^{-1}\mathbf{m}_j$$

- Moreover, if $P(\omega_j)=1/W$ and $\mathbf{C}_j=\mathbf{I}$:

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j$$

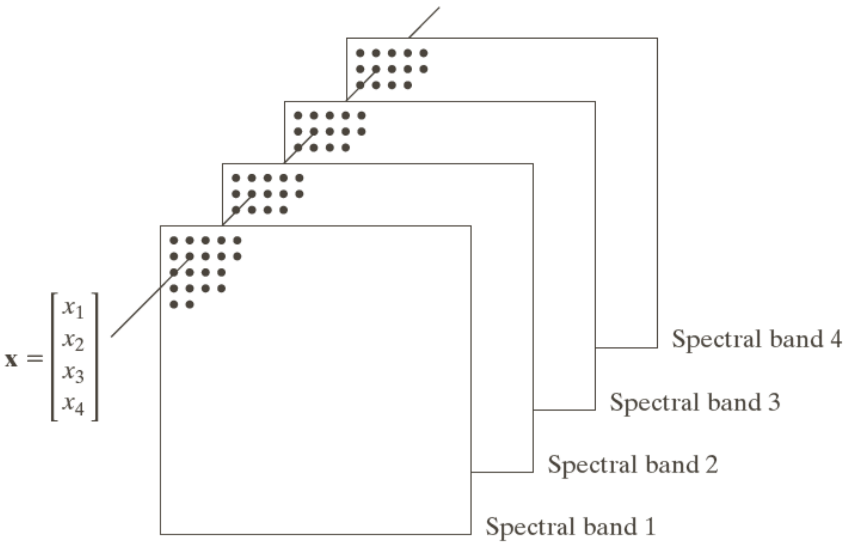
which is the minimum distance classifier decision function.

Bayes classifier for Gaussian pattern classes (cont.)

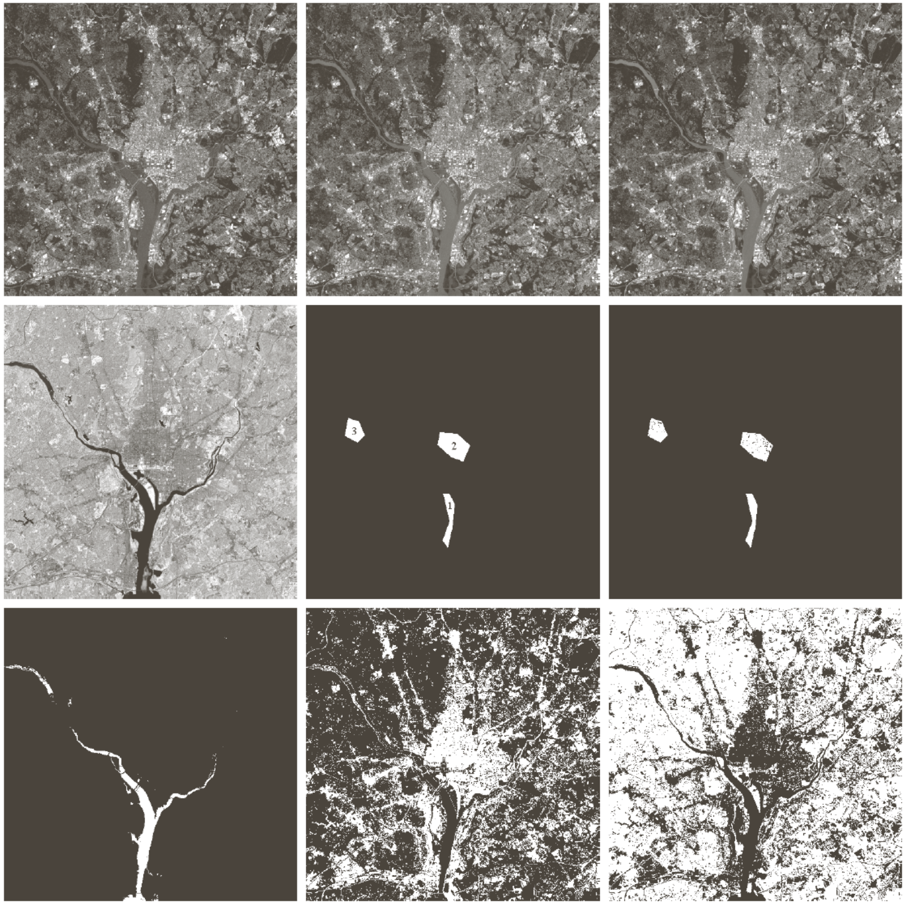
- The minimum distance classifier is optimum in the Bayes sense if:
  - The pattern classes are Gaussian.
  - All classes are equally to occur.
  - All covariance matrices are equal to (the same multiple of) the identity matrix.
- Gaussian pattern classes satisfying these conditions are spherical clouds (*hyperspheres*)
- The classifier establishes a *hyperplane* between every pair of classes.
  - It is the perpendicular bisector of the line segment joining the centers of the classes

Application to remotely sensed images

- 4-D vectors.
- Three classes
  - Water
  - Urban development
  - Vegetation
- Mean vectors and covariance matrices learnt from samples whose class is known.
  - Here, we will use samples from the image to learn the pdf parameters



Application to remotely sensed images (cont.)



a b c
d e f
g h i

FIGURE 12.13 Bayes classification of multispectral data. (a)–(d) Images in the visible blue, visible green, visible red, and near infrared wavelengths. (e) Mask showing sample regions of water (1), urban development (2), and vegetation (3). (f) Results of classification; the black dots denote points classified incorrectly. The other (white) points were classified correctly. (g) All image pixels classified as water (in white). (h) All image pixels classified as urban development (in white). (i) All image pixels classified as vegetations (in white).

Application to remotely sensed images (cont.)

Training Patterns						Independent Patterns					
		Classified into Class						Classified into Class			
Class	No. of Samples	1	2	3	% Correct	Class	No. of Samples	1	2	3	% Correct
1	484	482	2	0	99.6	1	483	478	3	2	98.9
2	933	0	885	48	94.9	2	932	0	880	52	94.4
3	483	0	19	464	96.1	3	482	0	16	466	96.7

Structural methods

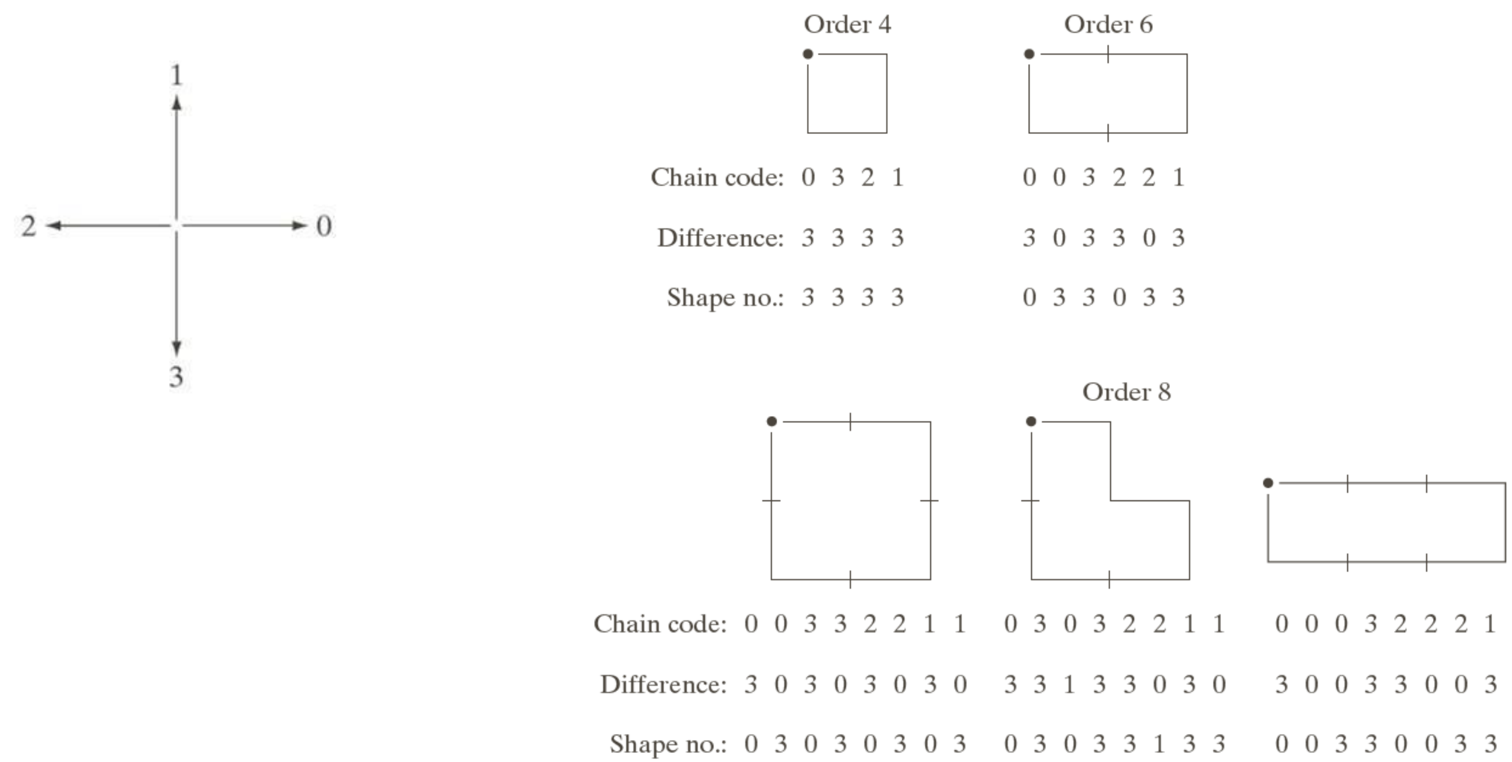
- Matching shape numbers
- String matching

Matching shape numbers

- The degree of similarity, k, between two shapes is defined as the largest order for which their shape numbers still coincide.
  - Reminder: The shape number of a boundary is the first difference of smallest magnitude of its chain code (invariance to rotation).
  - The order n of a shape number is defined as the number of digits in its representation.

Reminder: shape numbers

- Examples. All closed shapes of order n=4, 6 and 8.
- First differences are computed by treating the chain as a circular sequence.



Matching shape numbers (cont.)

- Let a and b denote two closed shapes which are represented by 4-directional chain codes and $s(a)$ and $s(b)$ their shape numbers.
- The shapes have a degree of similarity, k, if:

$$s_j(a) = s_j(b) \quad \text{for} \quad j = 4, 6, 8, \ldots, k$$
$$s_j(a) \neq s_j(b) \quad \text{for} \quad j = k+2, k+4, \ldots$$

- This means that the first k digits should be equal.
- The subscript indicates the order. For 4-directional chain codes, the minimum order for a closed boundary is 4.

Matching shape numbers (cont.)

- Alternatively, the distance between two shapes a and b is defined as the inverse of their degree of similarity:

$$D(a,b) = \frac{1}{k}$$

- It satisfies the properties:

$$D(a,b) \geq 0$$
$$D(a,b) = 0, \text{ iff } \; a = b$$
$$D(a,c) \leq \max[D(a,b),\; D(b,c)]$$

String matching

- Region boundaries a and b are code into strings denoted $a_1 a_2 a_3 \ldots a_n$ and $b_1 b_2 b_3 \ldots b_m$.
- Let p represent the number of matches between the two strings.
  - A match at the k-th position occurs if $a_k = b_k$.
- The number of symbols that do not match is:

$$q = \max(|a|,|b|) - p$$

- A simple measure of similarity is:

$$R = \frac{\text{number of matches}}{\text{number of mismatches}} = \frac{p}{q} = \frac{p}{\max(|a|,|b|) - p}$$