Biometric Authentication

Lecture 4

Biometrics Technology: Pattern Recognition

Outline

- Introduction: Pattern Recognition (PR)
- PR Models
- Simple Classifier

Introduction:
Pattern Recognition (PR)

Introduction: What is "Pattern"?

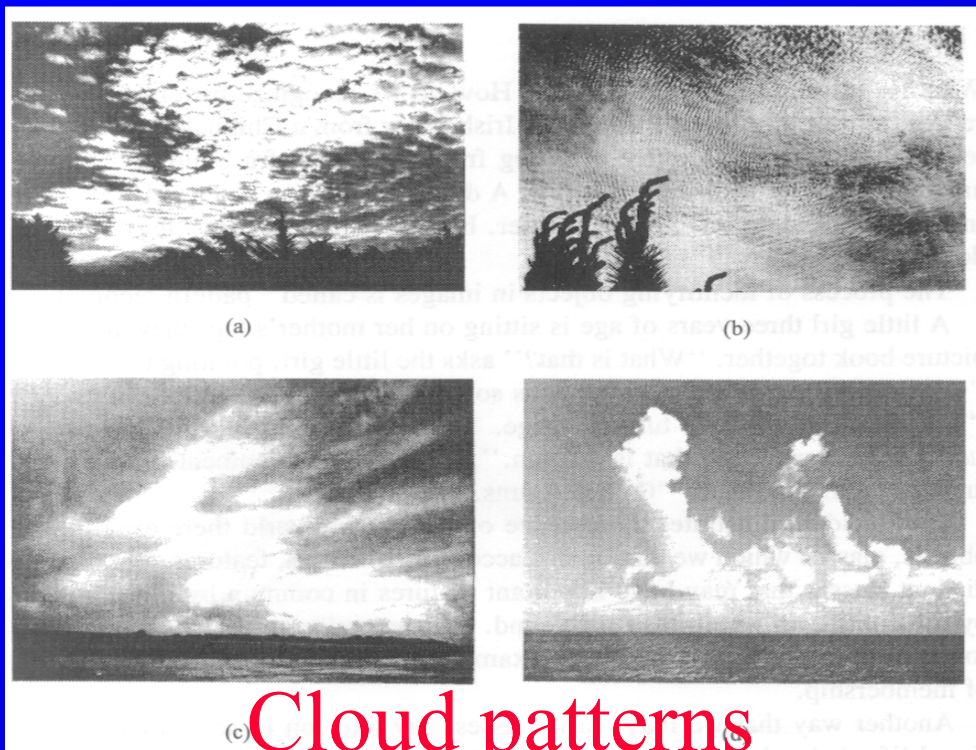
- A "pattern" is the form of representation of an objectively existed event or object. For instance, voice, image and character are patterns.

- There are many kinds of patterns -- visual patterns, temporal patterns, logical patterns. More broadly, *any natural and social phenomenon* may be considered as "Patterns".

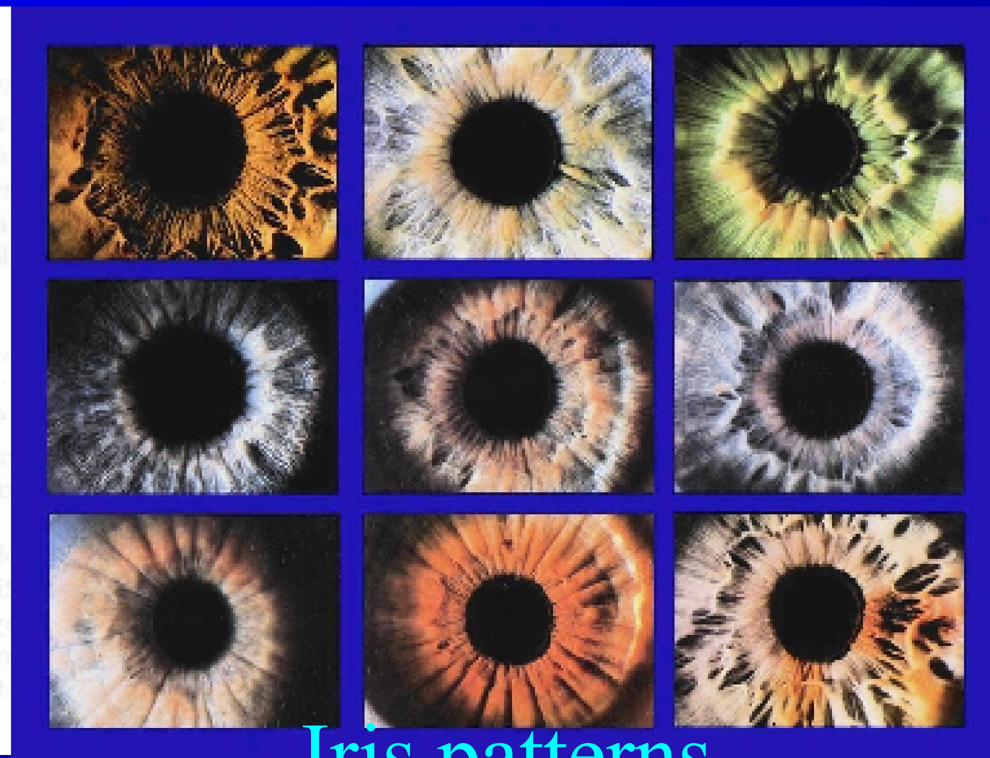
- We need to seek classification, recognition, or description of a pattern that is INVARIANT to some (known) changes or deviation in the pattern from the 'ideal' case

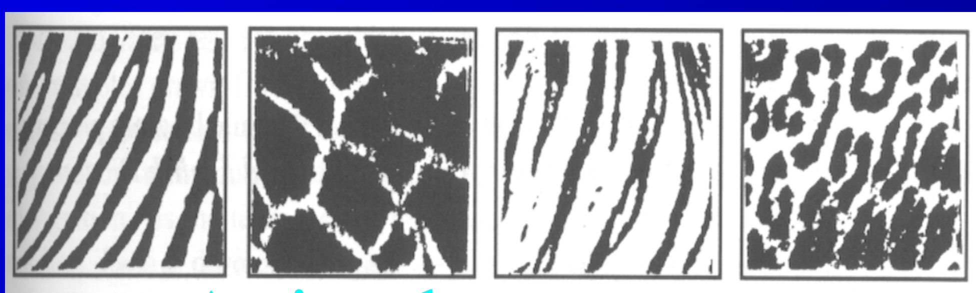
  $\rightarrow$ "Pattern" is **a set of measurements or observations, represented in vector or matrix notation**.

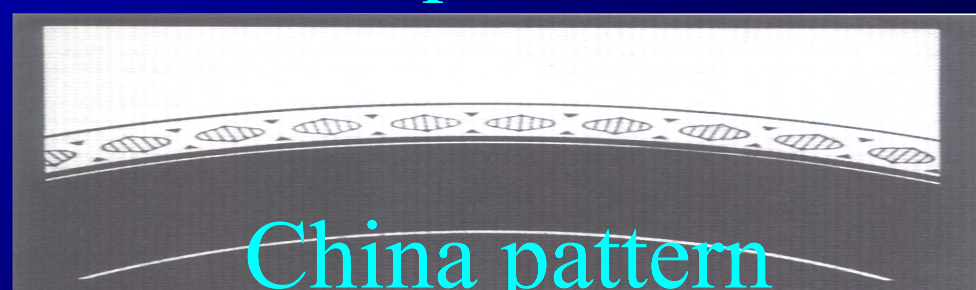
Introduction: What is “Pattern”?



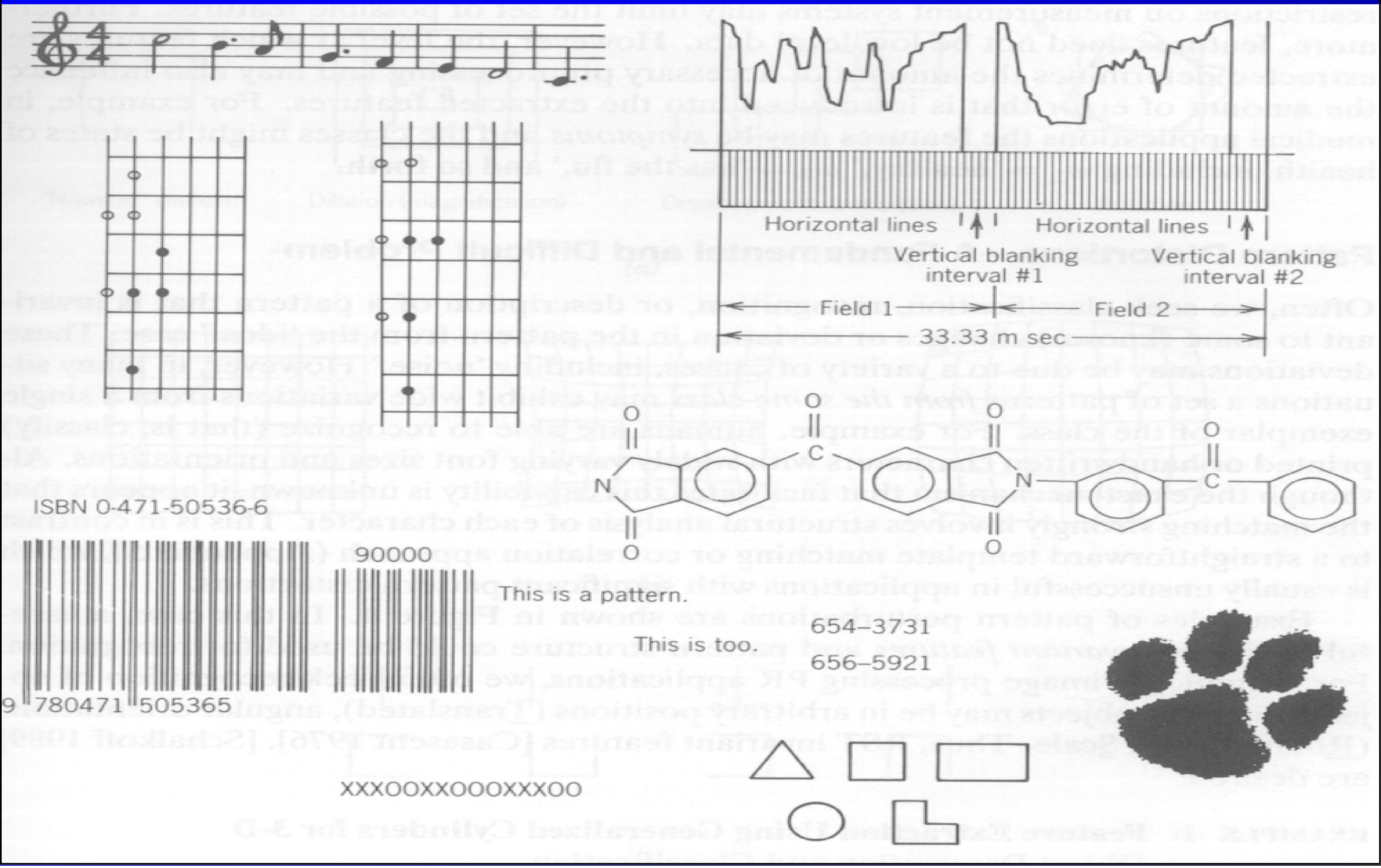
Cloud patterns

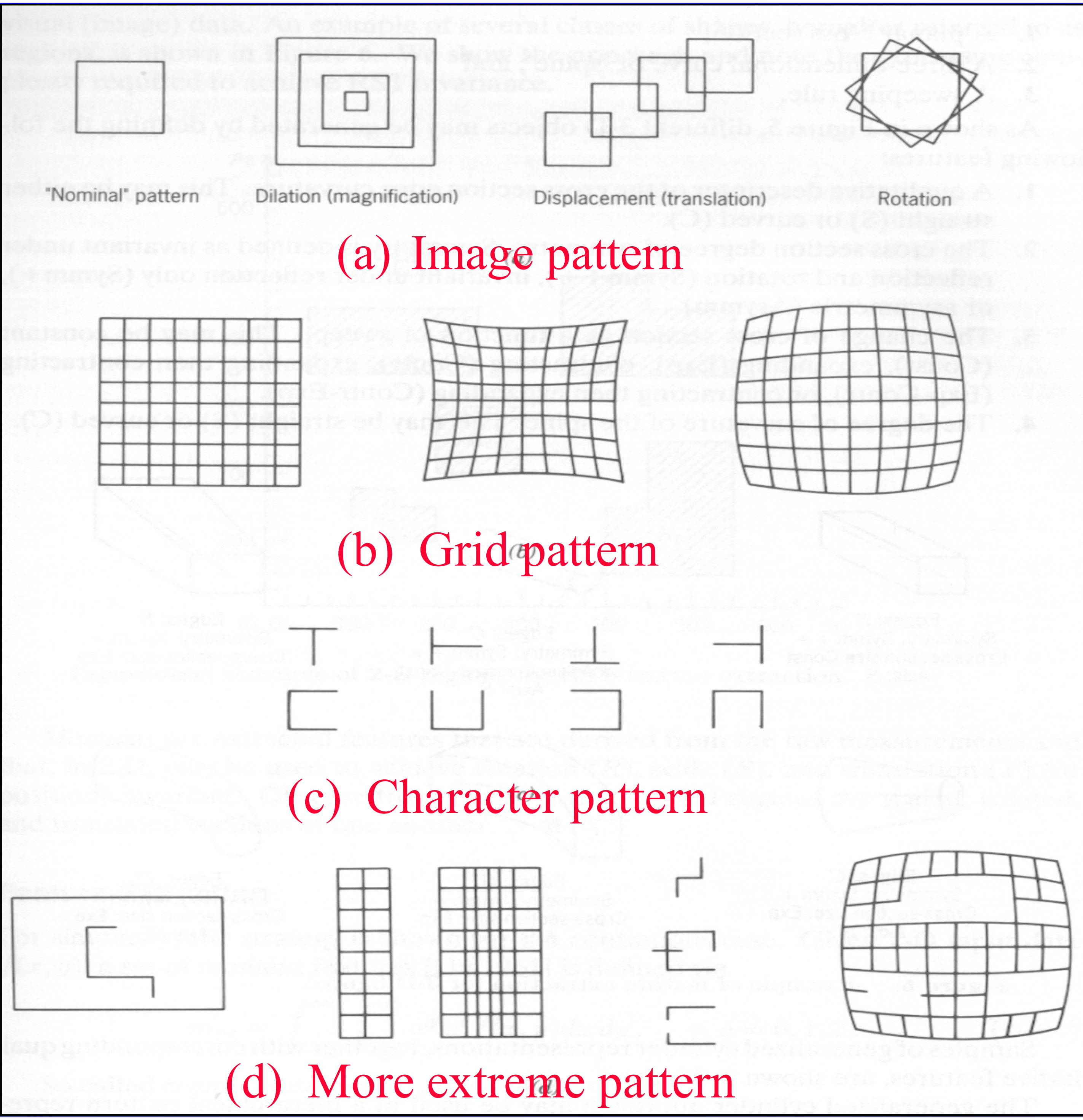

Iris patterns


Animal coat patterns


China pattern

Introduction: What is "Pattern"?





Pattern Distortion

- In many situations a set of patterns from the same class may exhibit wide variations from a single exemplar of the class.
- Invariant features: rotated, scaled, and translated invariant.

What is Pattern Recognition (PR)?

- A basic intelligent ability of human being or animal; Using a broad enough interpretation, we can find PR in every intelligent activity. For instance, you are able to recognize the number of classroom, which is the ability of *number recognition*, on the class you have to be able to understand what the teacher says and writes on the blackboard, this is the ability of *speech and character recognition*.
- From the system viewpoint, PR is an important component of intelligent systems.
- From the theoretical concept, PR is a mapping from **feature space** to **class space**.

Why we are interested in PR?

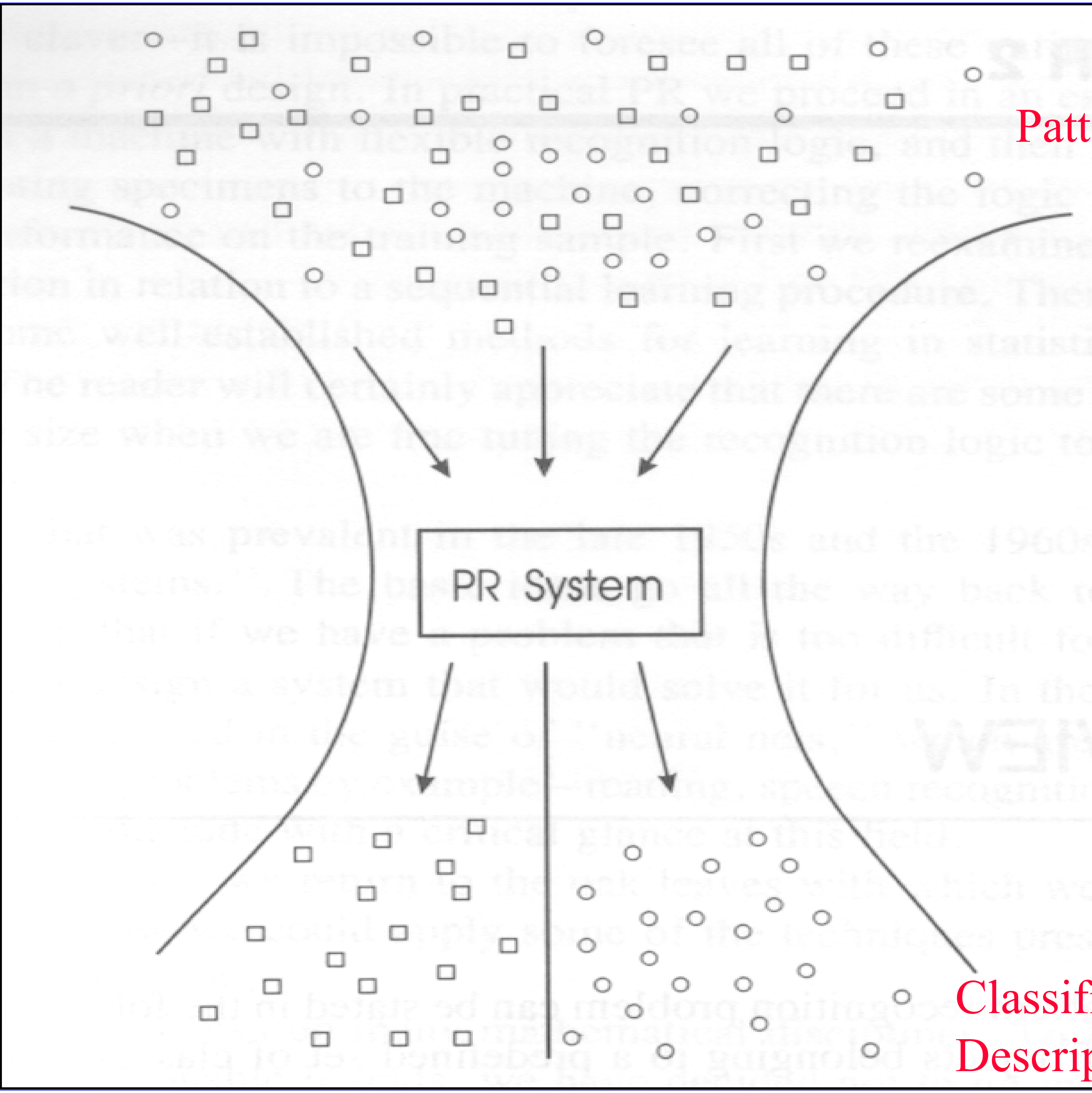
- Interest in the area of PR has been renewed recently due to emerging applications which are not only challenging but also computationally demanding. These applications include data mining (identifying a 'pattern', e.g., correlation, or an outlier in millions of multidimensional patterns), document classification (efficiently searching text documents), financial forecasting, and biometrics (personal identification based on various physical attributes such as face, etc.)
- The rapidly growing and available computing power, while enabling faster processing of huge data sets, has also facilitated the use of elaborate and diverse methods for data analysis and classification.

Why we are interested in PR?

- Demands on automatic PR systems are rising enormously due to the availability of large databases and stringent performance requirements (*speed, accuracy, and cost*).
- Recently scientists suggested a so-called affective computing which will give a computer the ability to recognize and express emotions, to respond intelligently to human emotion and to employ mechanism of emotion that contribute to rational decision making.

PR System

Pattern Data



Classification & Description

PR System

- The design of a PR system essentially involves the following three aspects :

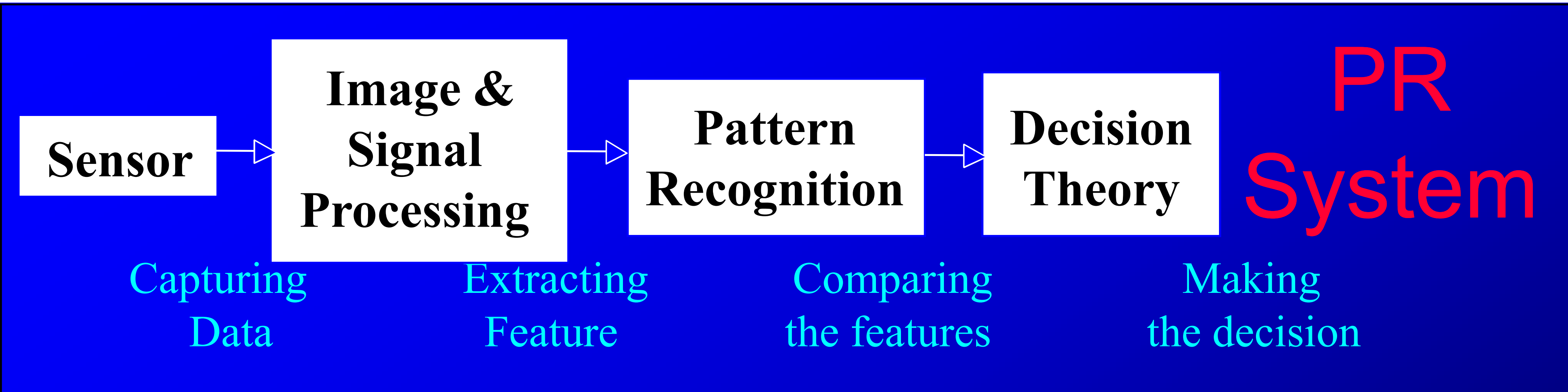
  1) data acquisition and preprocessing;

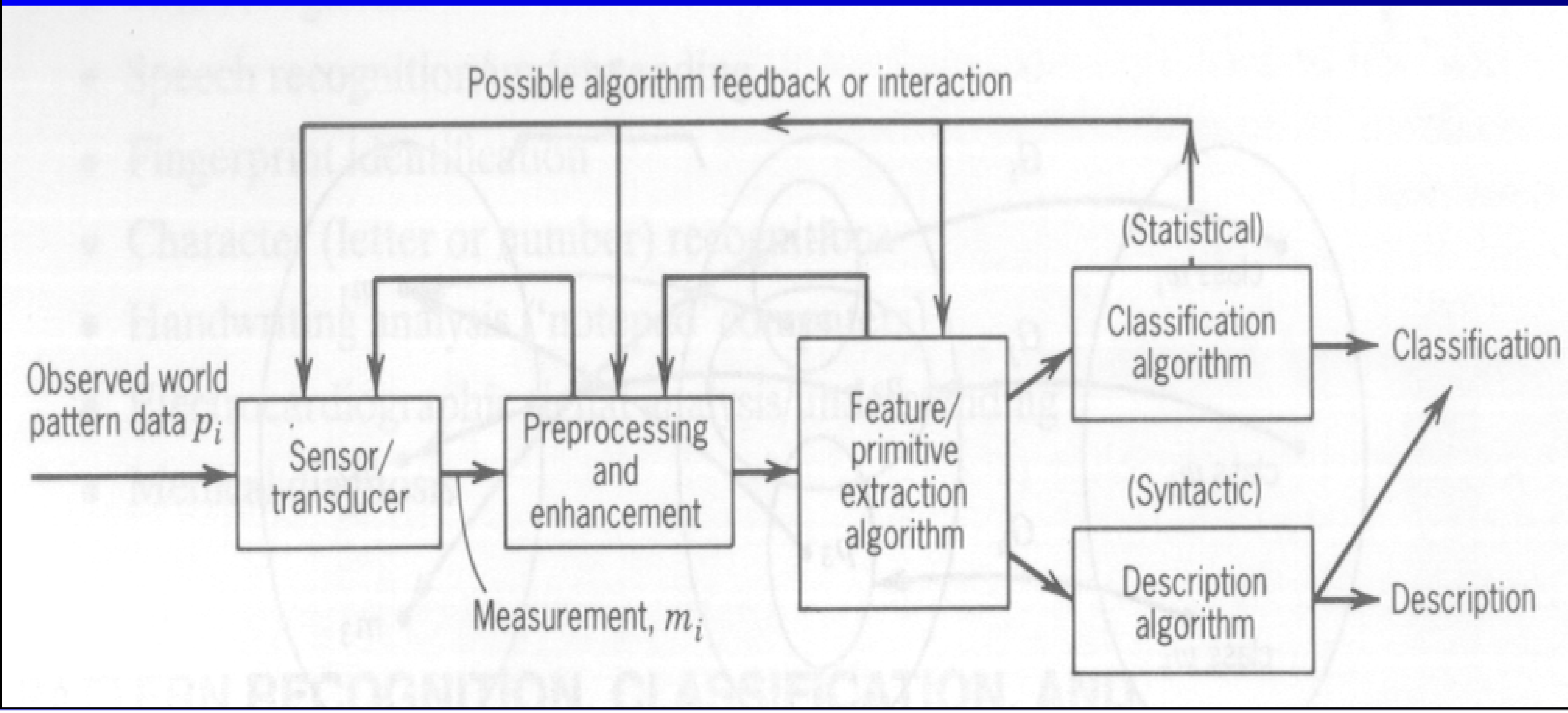
  2) data representation;

  3) decision making.

  The problem domain dictates the *choice of sensor*, *preprocessing technique, representation scheme,* and *the decision making model*

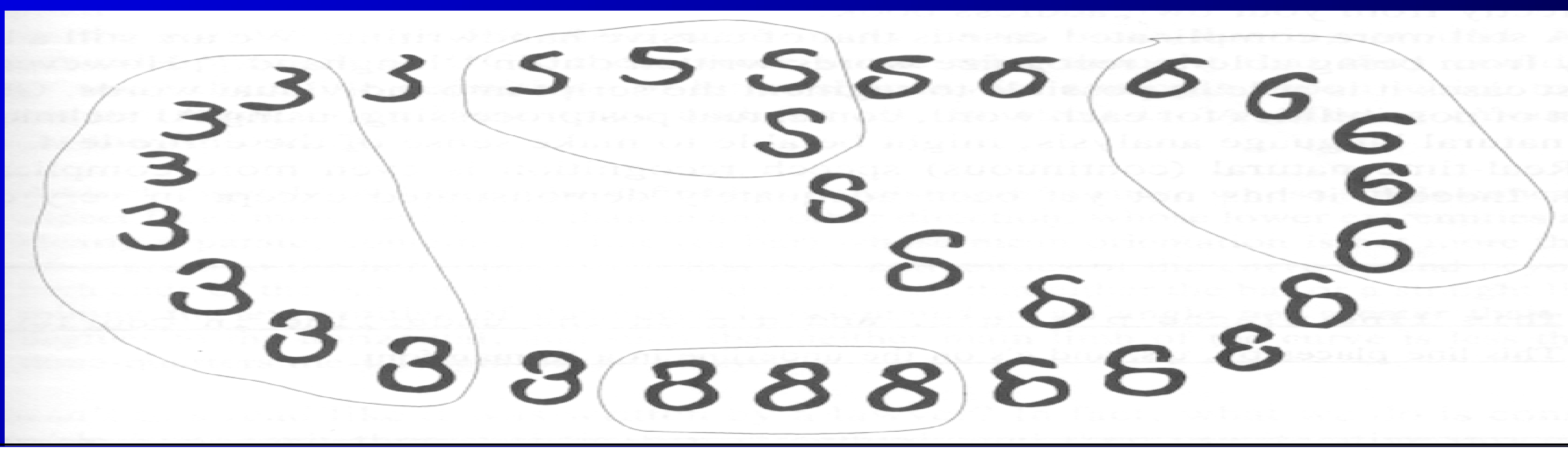
- It is generally agreed that a well-defined and sufficiently constrained recognition problem (small intraclass variations and large interclass variations) will lead to a compact pattern representation and a simple decision making strategy.


Sensor
Image & Signal Processing
Pattern Recognition
Decision Theory
PR System
Capturing Data
Extracting Feature
Comparing the features
Making the decision


Possible algorithm feedback or interaction
Observed world pattern data p_i
Sensor/ transducer
Preprocessing and enhancement
Feature/ primitive extraction algorithm
(Statistical)
Classification algorithm
(Syntactic)
Description algorithm
Classification
Description
Measurement, m_i

Definitions

- *Classification* - Assigns input data into one of *c* pre-specified classes based on extraction of significant features and the processing or analysis of these features. It is common to resort to probabilistic or grammatical models in classification. A *Classifier* partitions feature space into class-labeled *decision regions*.
- *Recognition* is the ability to classify. Often we formulate PR with a *c + 1st* class, corresponding to the 'unclassifiable' or 'can't decide' class.



Definitions (2)

- *Pattern class* - A set of patterns (hopefully sharing some common attributes) known to originate from the same source. The key in many PR applications is to identify suitable attributes (e.g., features) and to form a good measure of similarity and an associating matching process.
- *Preprocessing* is the filtering or transforming of the raw input data to aid computational feasibility and feature extraction and minimize noise.
- *Description* is an alternative to classification where a structural description of the input pattern is desired. It is common to resort to linguistic or structural models in description.

Pattern Extraction: GC Example

- Feature extraction using Generalized Cylinders (GC) for 3-D object description and classification
- The basis of GC models is the concept of a ‘swept volume’ of 2-D area along a 3-D trajectory.
- A GC is a solid whose axis is a 3-D space curve. Usually the axis is perpendicular to the cross section. For example, the typical cylinder or ‘can’ may be described by sweeping a circle along a line.
- We characterize a generalized cylinder (GC) by three parameters:

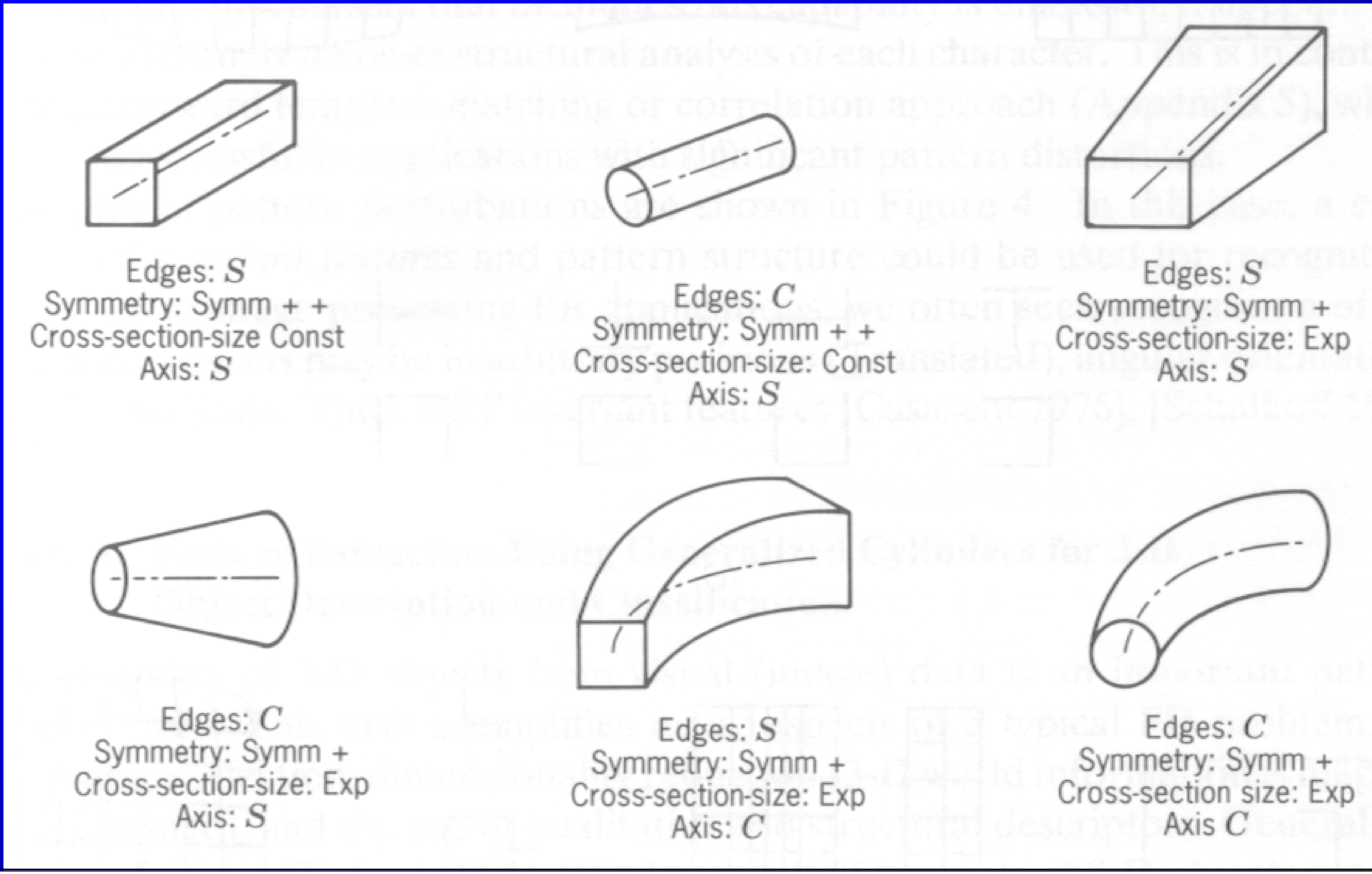
  1. A planar cross section;
  2. A three-dimensional curve or ‘spine’;
  3. A sweeping rule.

Pattern Extraction: 3-D Example

- 3-D objects, defining the following features

1. A qualitative descriptor of the cross section edge curvature. This may either straight (S) or curved (C).
2. The cross section degree of symmetry, defined as :
   -- Invariant under reflection and rotation (Symm ++)
   -- Invariant under reflection only (Symm +)
   -- Asymmetric (Asymm).
3. The change of cross section as a function of sweep, defined as
   -- Constant (Const)
   -- Expanding (Exp)
   -- Contracting (Contr)
   -- Contracting then expanding (Contr-Exp)
   -- Expanding then contracting (Exp-Contr)
4. The degree of curvature of the spine. This may be straight (S) or curved (C).

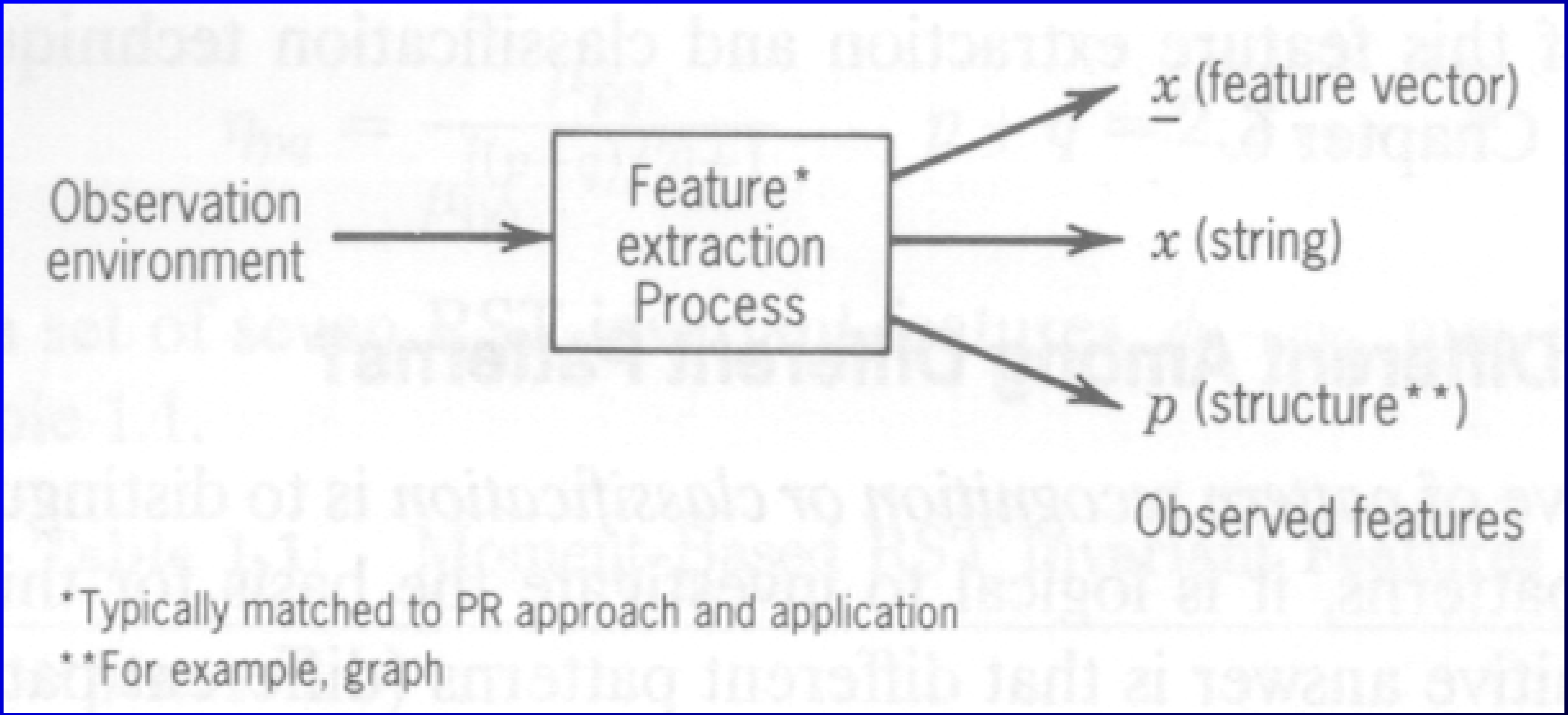
Pattern Extraction: 3-D Example



Pattern Vector & Pattern Space

- The objective of PR or CLASSIFICATION is to distinguish between different types of patterns.
- Different patterns are composed of different features or features with different numerical values. Features are arranged in a *d* dimensional *feature vector,* denoted x which yields a multidimensional measurement *space* or *feature space,* R^d.
- But, this is not generally true. Consider a newspaper picture, which consists of an array of dots, each being ‘white’ or ‘black’. If we use these dots as features, then we may discover that all pictures are made up of the same features. *How can we distinguish between a picture of a car and a picture of a boat based on these same features?* In this case, we may resort to taking advantage of the *spatial arrangements of these features*.

Feature Extraction



Pattern data → Feature

PR Models

Pattern Recognition Models

We will view pattern recognition as **classification** -- assigning an input to a category. By restricting the problem in this way, we will develop some useful general techniques that will satisfy our needs for this course.

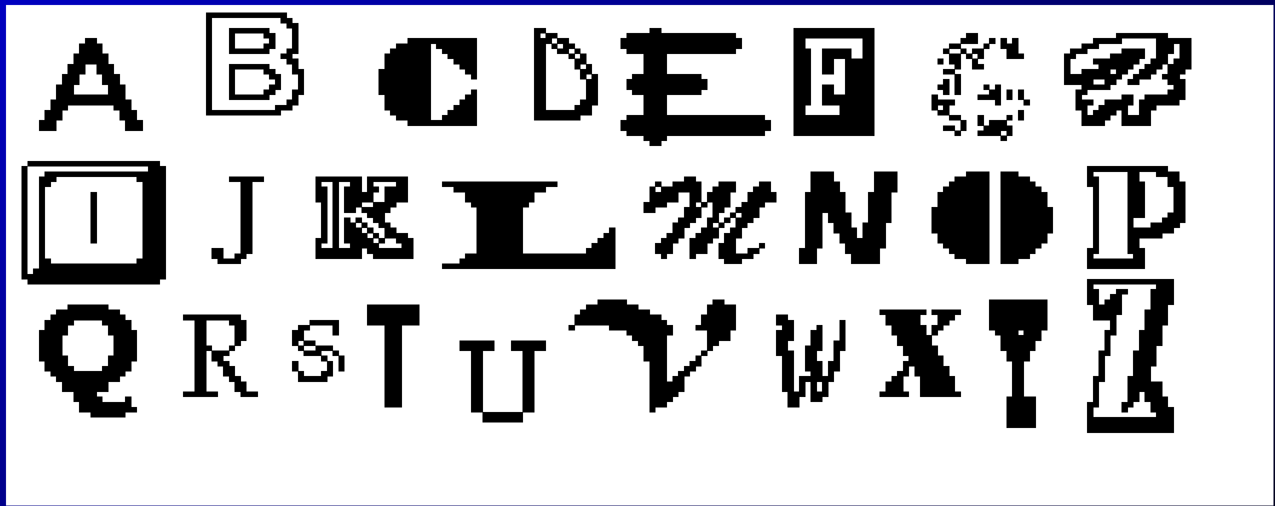
In this section, we focus on the following basic concepts:

- Classification;
- Features;
- Feature vectors;
- Standard model

Classification

We will view pattern recognition as **classification** -- assigning an input to a category. Suppose that we are working with the 26 letters of the Roman alphabet. Then we can say that the pattern recognition problem is one of assigning the input to one of 26 classes.

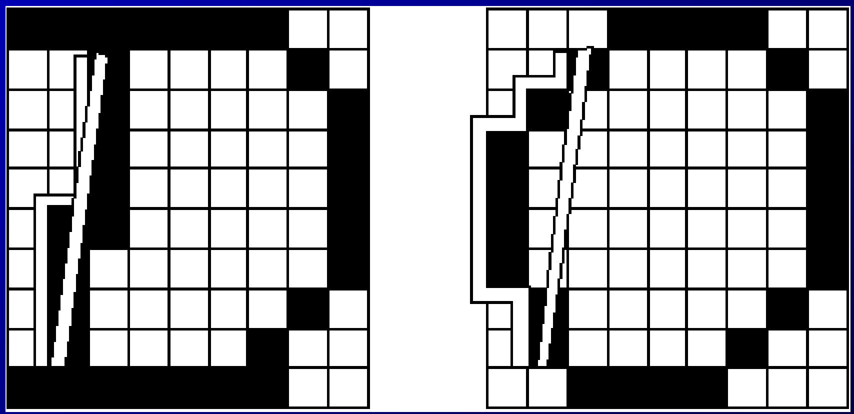
Suppose further that we have used a camera to digitize the visual input, and we have isolated a single character as an array of brightness values. What a computer classify this data is to compare the input with a standard pattern for each class, and to choose the class that matches best.



Features

One way to classify an object or event is to measure its characteristic **properties** or **features**. For example, to classify a printed letter, it might be helpful to know its area and its perimeter.

Some properties might be particularly sensitive to important small differences. For example, to distinguish the letter "D" from the letter "O", we might measure the straightness of the left side, perhaps by computing the ratio of the straight -line distance to the arc length.



Clearly, the features that are needed depend on the specific problem that one wants to solve, and designing a good set of features is more of an art than a science.

What is Features?

- Broadly speaking, FEATURES are any extractable measurement used.
  - Features may be symbolic, e.g., color
  - Features may be numerical, e.g., weight
  - Features may be both symbolic and numerical
  - Features may also be HIGHER LEVEL ENTITIES, e.g., geometric descriptors of either an image region or a 3-D object appearing in the image.
- Any features can not be used for everything. For a given event or object, its special measurements extracted should be different from others.

How to get Features?

- From applying a *feature extraction algorithm or operator* to the input data (such as 1-D data: voice; 2-D data: image; 3-D data: graphics, etc.)
- **Notice**
  1. Significant computational effort may be required in feature extraction;
  2. The extracted features may contain errors or 'noise';
  3. Features may be represented by continuous, discrete, or discrete-binary variables.
- Not all features are useful for a special problem.

Feature Selection

- It is the process of choosing input to PR system and involves judgment.
- It is important that the extracted features be relevant to the PR task at hand.
- It has to be addressed at the outset of any PR system design.
- To choose and to extract features that :

  1. are computationally feasible;
  2. lead to 'good' PR system success;
  3. reduce the problem data (e.g., raw measurements) into a manageable amount of information without discarding valuable (or vital) information

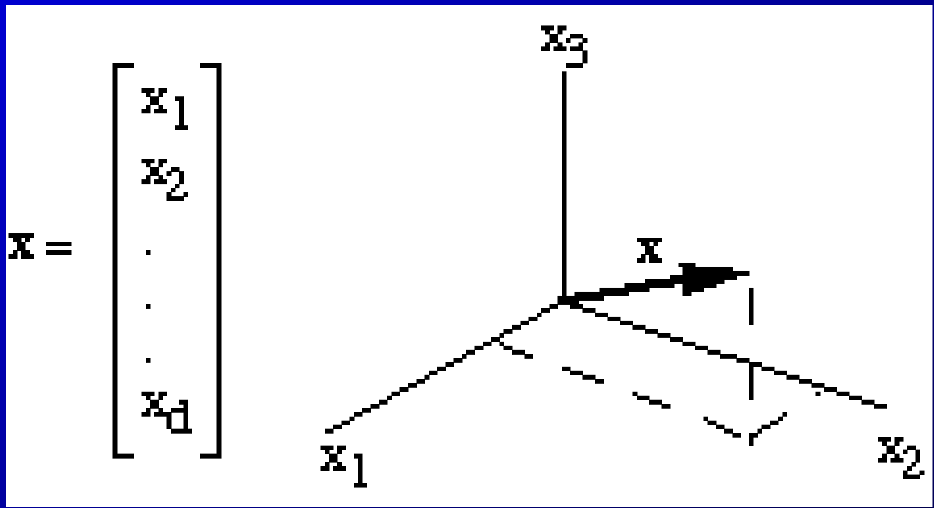
Feature Selection

- In some cases there are mathematical tools that help in feature selection. In other cases simulation may help to choose appropriate possible features. *Restrictions on measurement systems may limit the set of features*. The level at which features are extracted determines the amount of necessary preprocessing and may influence the amount of error that is introduced into the extracted features
- The measurements may be a 2-Dimensional image, a drawing, a waveform, a set of measurements, a temporal or spatial history (sequence) of events, the state of a system, the arrangement of a set of objects, and so forth.

Feature Vectors

It frequently happens that we can measure a fixed set of *d* features for any object or event that we want to classify. For example,

x1 = area
x2 = perimeter
...
xd = arc_length/straight_line_distance

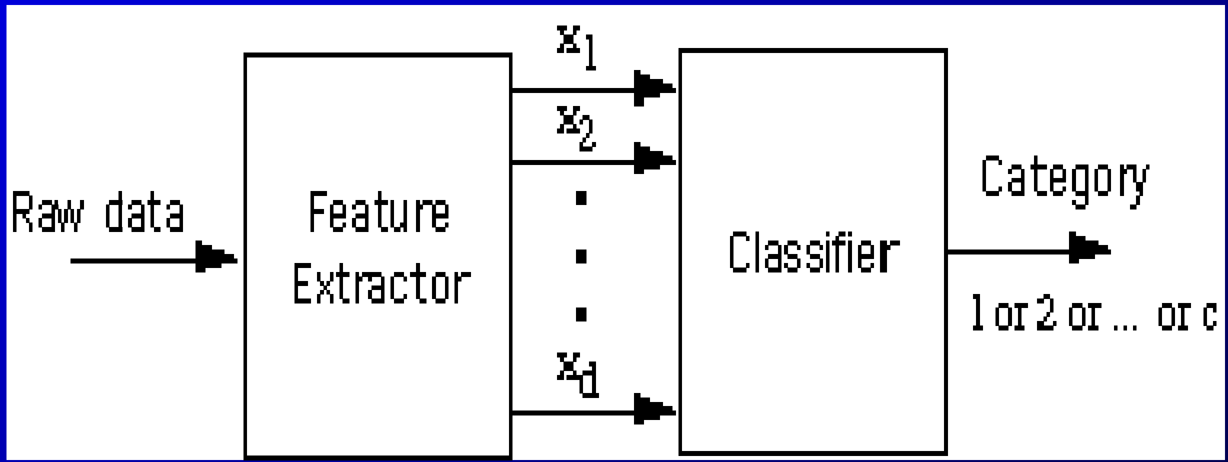


In this case, we can think of our feature set as a **feature vector x**, where **x** is the d-dimensional column vector.

Equivalently, we can think of **x** as being a point in a d-dimensional **feature space**. By this process of feature measurement, we can represent an object or event abstractly as a point in feature space.

Standard Model

Feature extractor processes the raw data to determine the numerical values for a set of *d* features x1, x2, ..., xd, which comprise the components of a feature vector **x**. **Classifier** receives **x** and assigns it to one of c categories, *Class 1, ..., Class c*.



The design of the feature extractor is very problem dependent. The ideal feature extractor would produce the same feature vector **x** for all patterns in the same class, and different feature vectors for patterns in different classes. In practice, different inputs to the feature extractor will always produce different feature vectors, but we hope that the within-class Variability is small relative to the between-class variability.

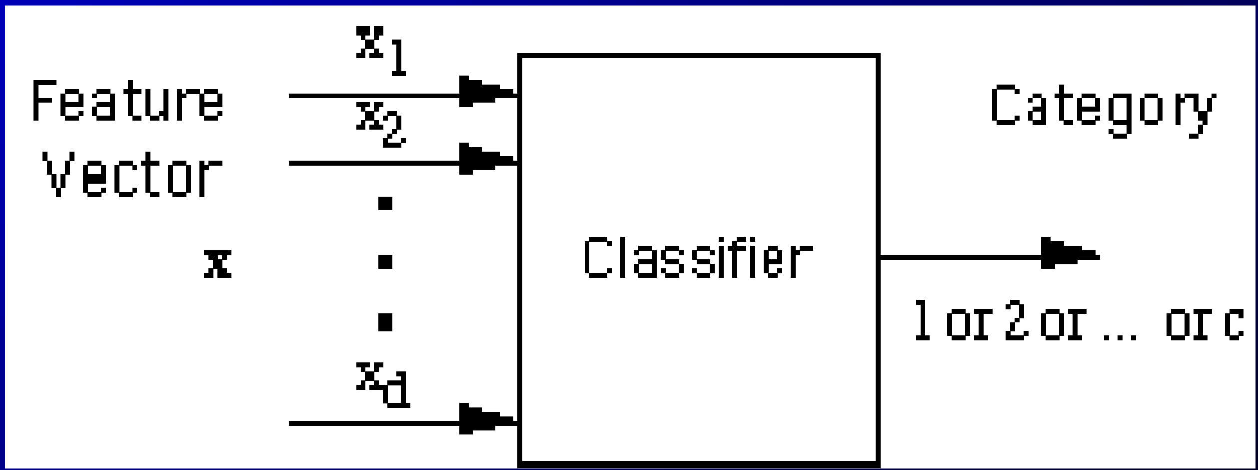
Simple Classifier

Simple Classifiers

There are at least two ways to approach the design of a classifier:

- Hypothesize a plausible solution and adjust it to fit the problem
- Create a mathematical model of the problem and derive an optimal classifier

The first method is more intuitive, is frequently used in practice, and is the approach that we shall take. We start with a very simple solution, analyze its characteristics, identify its weaknesses, and complicate it only as necessary. In this section we cover the following topics: Template matching; Minimum-distance classifiers; Metrics; Inner products; Linear discriminants; and Decision boundaries

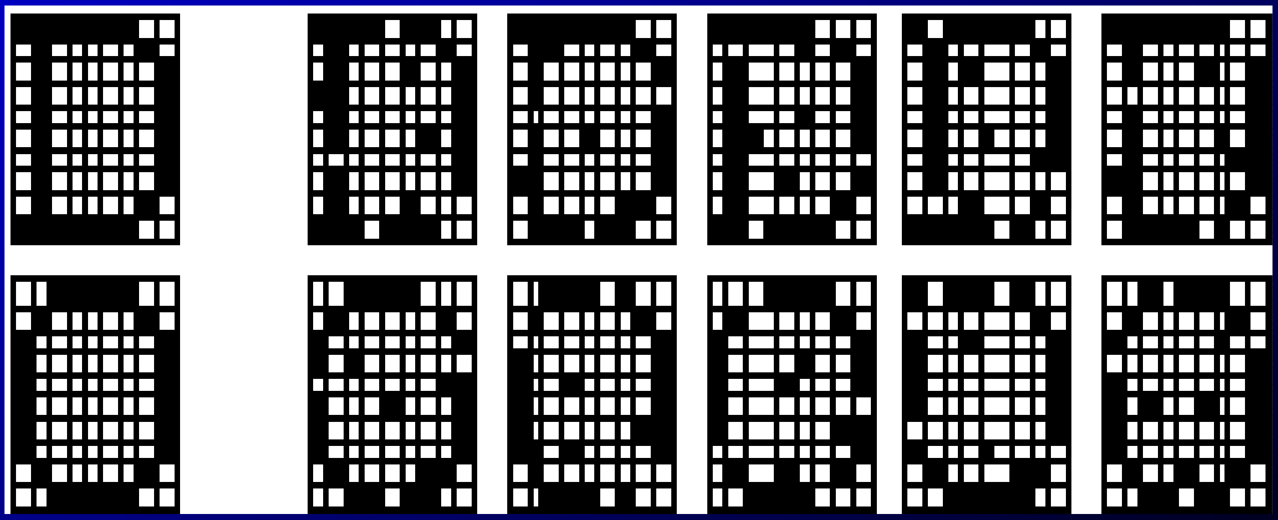


Template Matching

Template matching is a natural approach to pattern classification. For example, consider the noisy "D"'s and "O"'s. The noise-free versions can be used as **templates**. To classify one of the noisy examples, simply compare it to the two templates. Two ways can be done:

1. Count the number of agreements (black matching black and white matching white). Pick the class that has the maximum number of agreements. This is a **maximum correlation** approach.

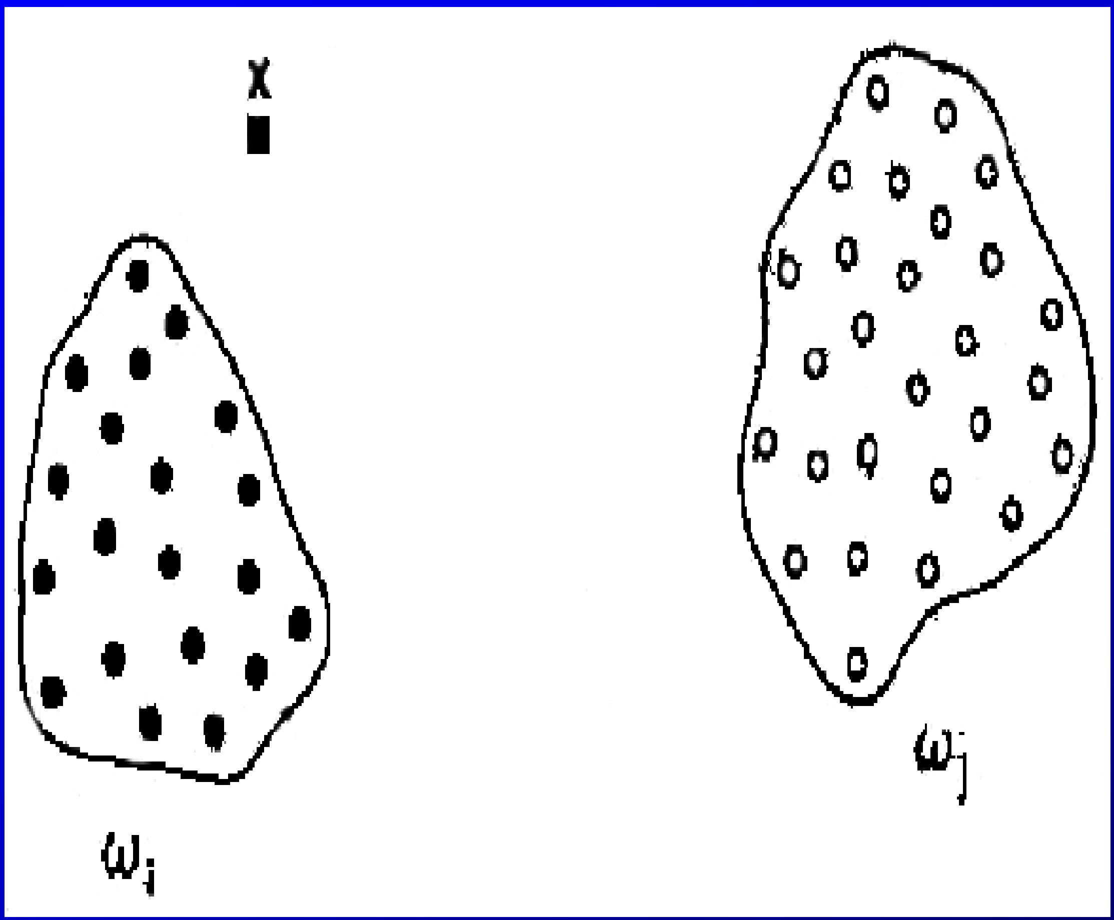
2. Count the number of disagreements (black where white should be or white where black should be). Pick the class with the minimum number of disagreements. This is a **minimum error** approach.



Pattern Classification by Distance Function

- Pattern classification, or decision, process is intended to carry out the final classification of the unknown objects.
- The motivation for using distance functions as a classification tool follows naturally from the fact that the most obvious way of establishing a measure of similarity between pattern vectors, which we also consider as points in Euclidean space, is by determining their proximity.
- Based on proximity concept, the method is suitable only when the pattern classes tend to have clustering properties. However, some patterns are classifiable by proximity concept; but, some are not easily.

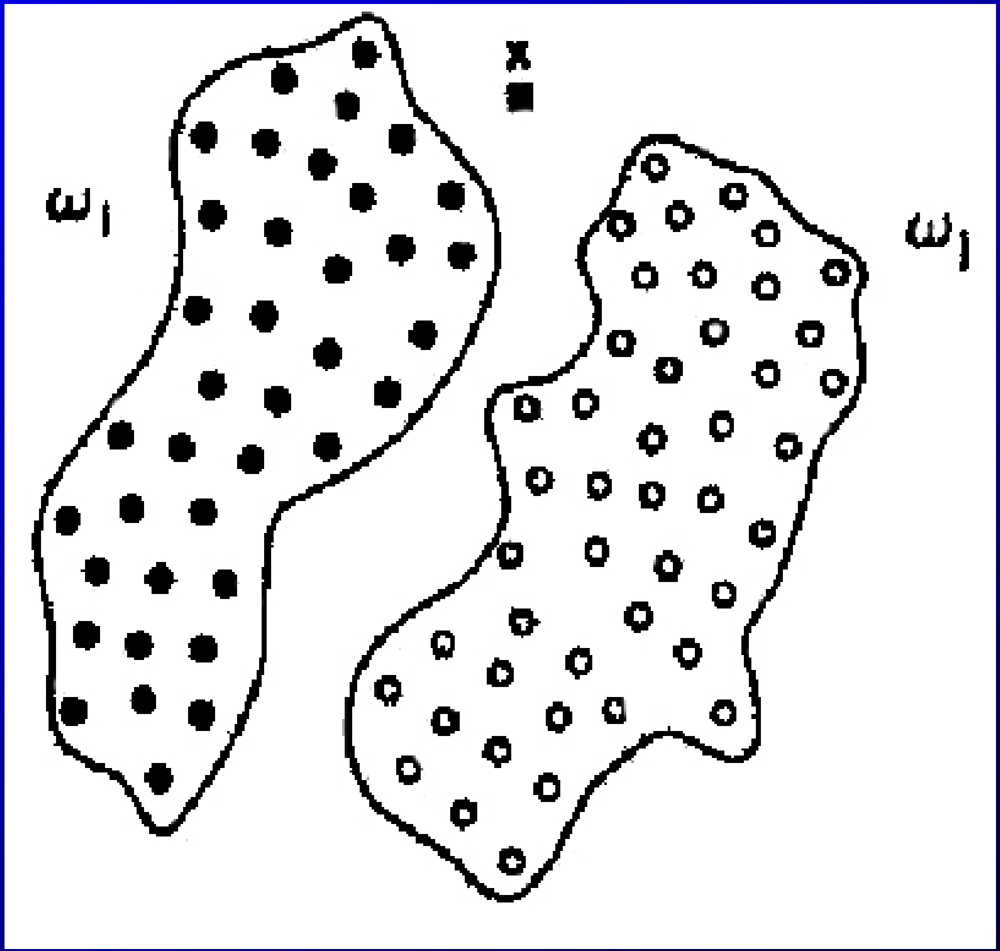
Pattern Classification by Distance Function



- In this case, we may intuitively arrive at the conclusion that X belongs to class ϖ_i solely on the basis that it is closer to the patterns of this class.

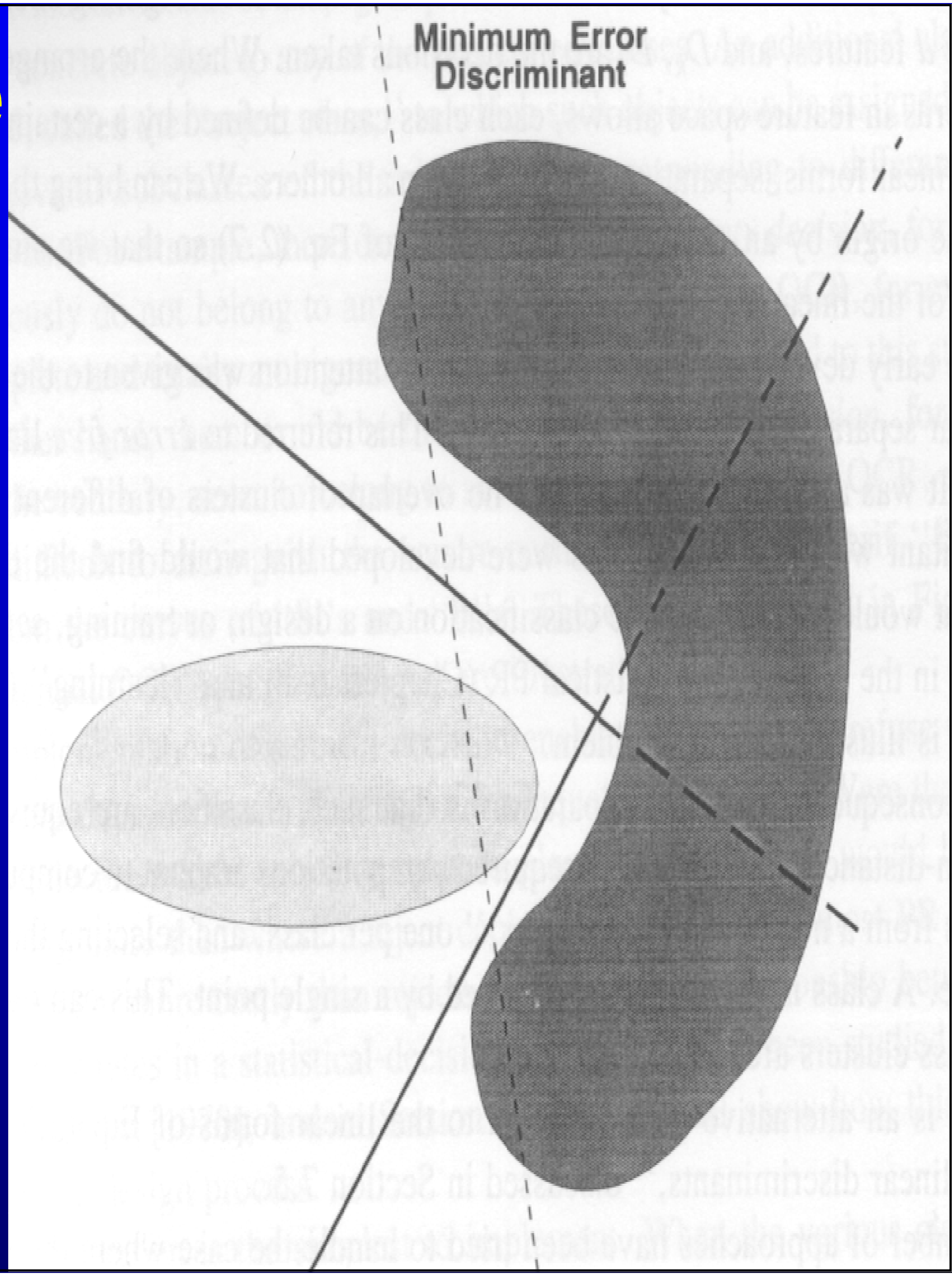
Pattern Classification by Distance Function

- But in this case, it is difficult to arrive at a justification for classifying into either class based on a measure of the proximity of this pattern to a class.



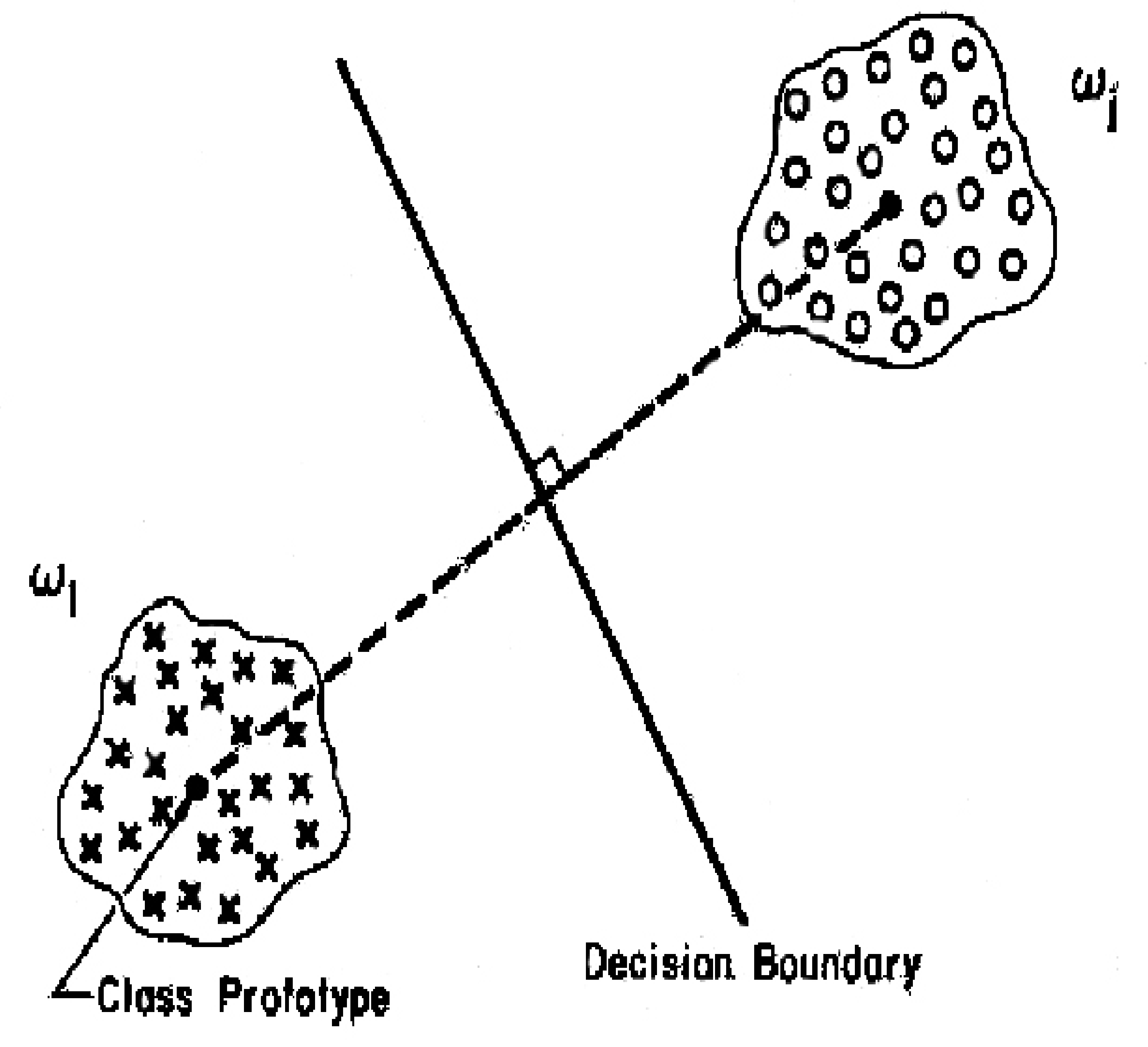
Minimum-Distance Classification

- Linear separation is the simplest to implement in pattern classification. It is referred to error-free linear separation, which is assumed that there are no overlap of clusters of different classes.
- Our goal is to find the discriminates that would yield error-free classification on input data.



Minimum-Distance Classification

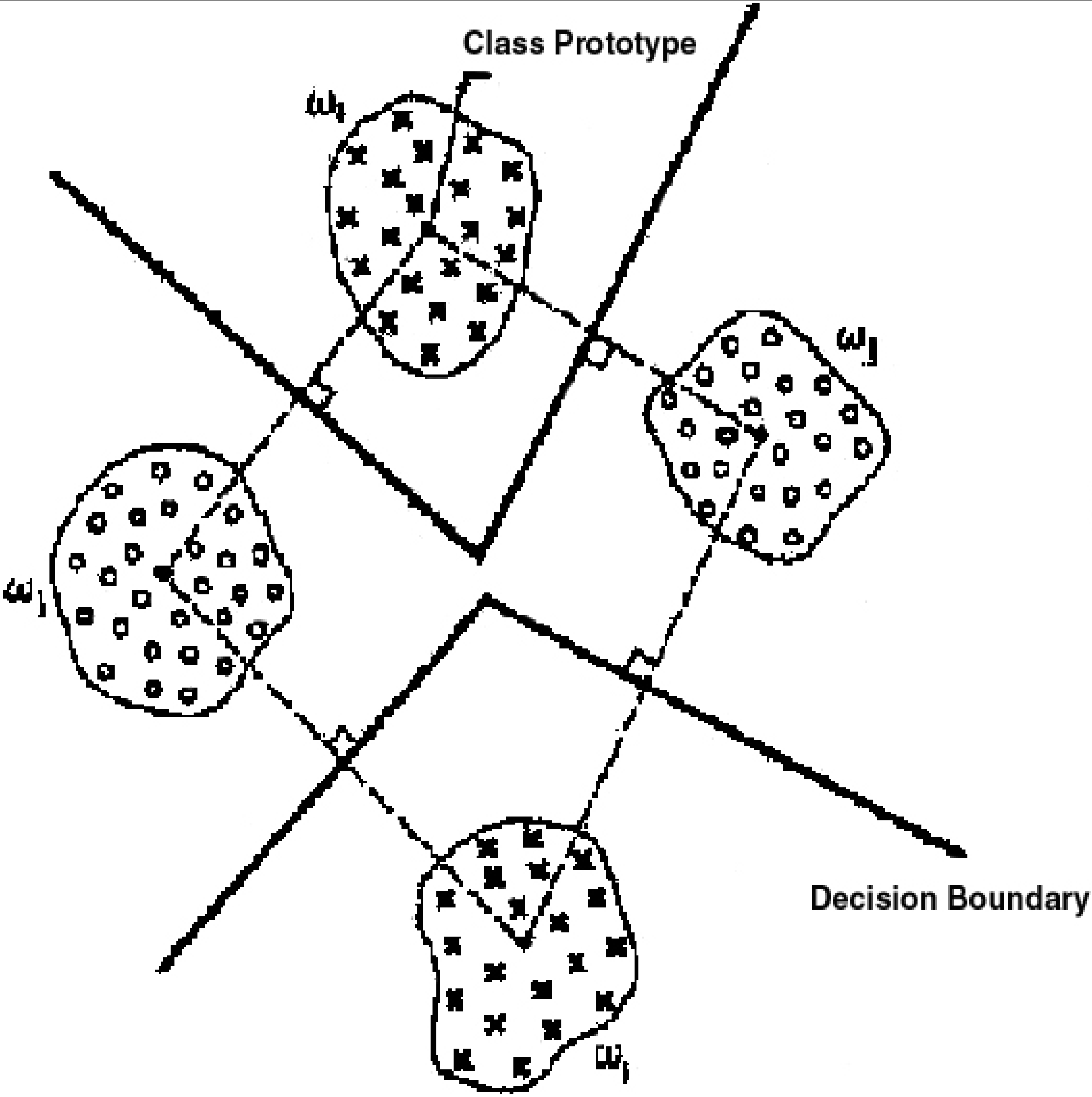
- One consequence of linear separation is that such classifiers are equivalent to minimum-distance classifier. The required computations amount to computing the distances from a number of "prototype," one per class, and selecting the nearest prototype.
- Since a minimum-distance classifier categorizes a pattern on the basis of the closest match between the pattern and the respective class prototypes, this approach is also known as ***correlation or cluster matching*** **.**
- There are two kinds of prototypes: Single Prototype and Multi Prototypes



Single Prototype

- Decision boundary of two classes, characterized by single prototypes.

Multi-Prototypes

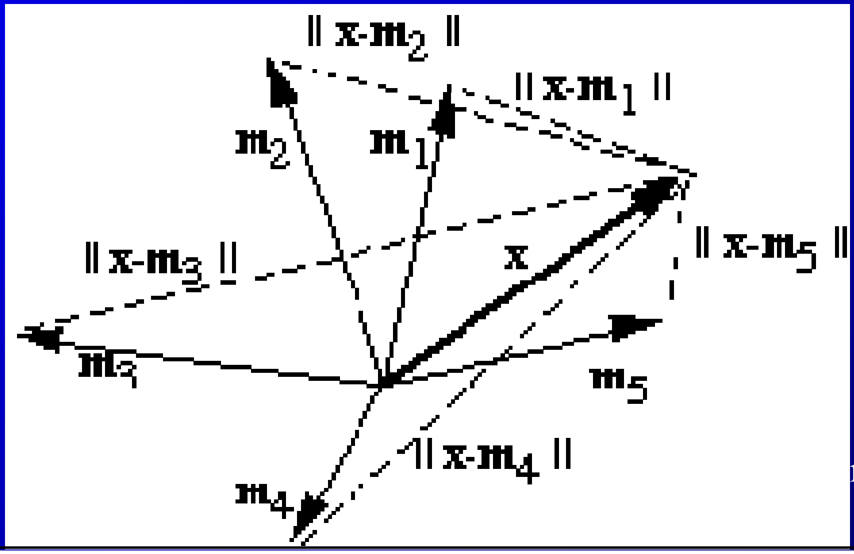


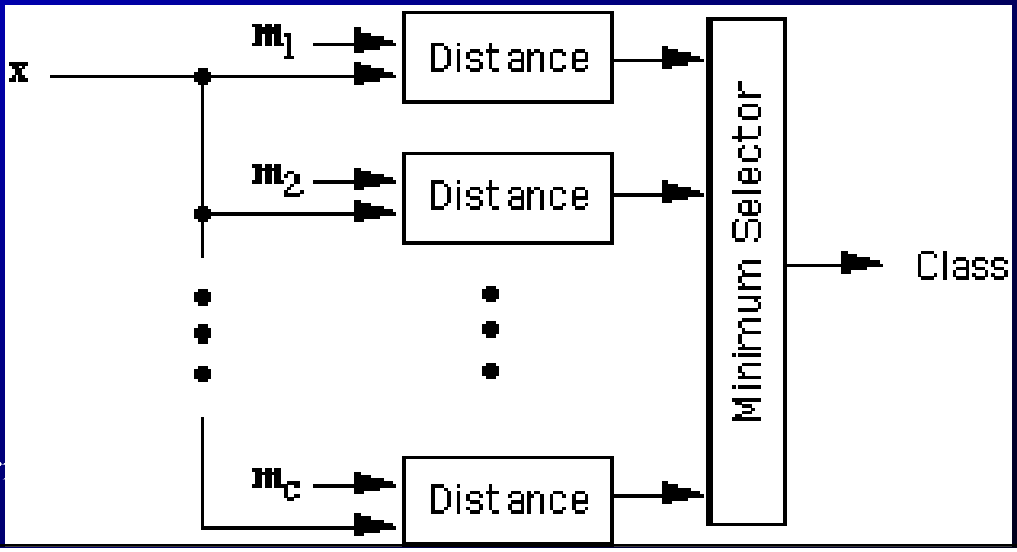
- Piecewise-linear decision boundaries for two classes, each of which is characterized by two prototypes.

Minimum-Distance Classifiers

Template matching can easily be expressed mathematically. Let **x** be the feature vector for the unknown input, and let **m**1, **m**2, ..., **mc** be templates (i.e., perfect, noise-free feature vectors) for the c classes. Then the error in matching **x** against **mk** is given by || **x** - **mk** ||.

Here || **u** || is called the **norm** of the vector **u**. A minimum-error classifier computes || **x** - **mk** || for k = 1 to c and chooses the class for which this error is minimum. Since || **x** - **mk** || is also the distance from **x** to **mk**, we call this a **minimum-distance** classifier. Clearly, a template matching system is a minimum-distance classifier.





Metrics

There is more than one way to define the norm || **u** ||, and these correspond to different ways of measuring distance, i.e., to different **metrics**. Two of the most common are

Euclidean metric: || **u** || = sqrt(u12 + u22 + ... + ud2)
Manhattan (or taxicab) metric: || **u** || = |u1| + |u2| + ... + |ud|

In our template-matching example of classifying characters by counting the no. of disagreements, we were implicitly using a Manhattan metric.

- Contours of constant Euclidean distance are circles (or spheres)
- Contours of constant Manhattan distance are squares (or boxes)
- Contours of constant Mahalanobis distance are ellipses (or ellipsoids)

Linear Discriminants (1)

Recall that when we use a minimum-distance classifier to classify a feature vector **x**, we measure the distance from **x** to the templates **m**1, **m**2, ..., **m**c and assign **x** to the class of the nearest template. Using the inner product to express the Euclidean distance from **x** to **m**k, we can write

$$\| \mathbf{x} - \mathbf{m}_k \|^2 = (\mathbf{x} - \mathbf{m}_k)' (\mathbf{x} - \mathbf{m}_k)$$

$$= \mathbf{x}'\mathbf{x} - \mathbf{m}_k'\mathbf{x} - \mathbf{x}'\mathbf{m}_k + \mathbf{m}_k'\mathbf{m}_k$$

$$= -2 \, [\, \mathbf{m}_k'\mathbf{x} - 0.5 \, \mathbf{m}_k \mathbf{m}_k \,] + \mathbf{x}'\mathbf{x}$$

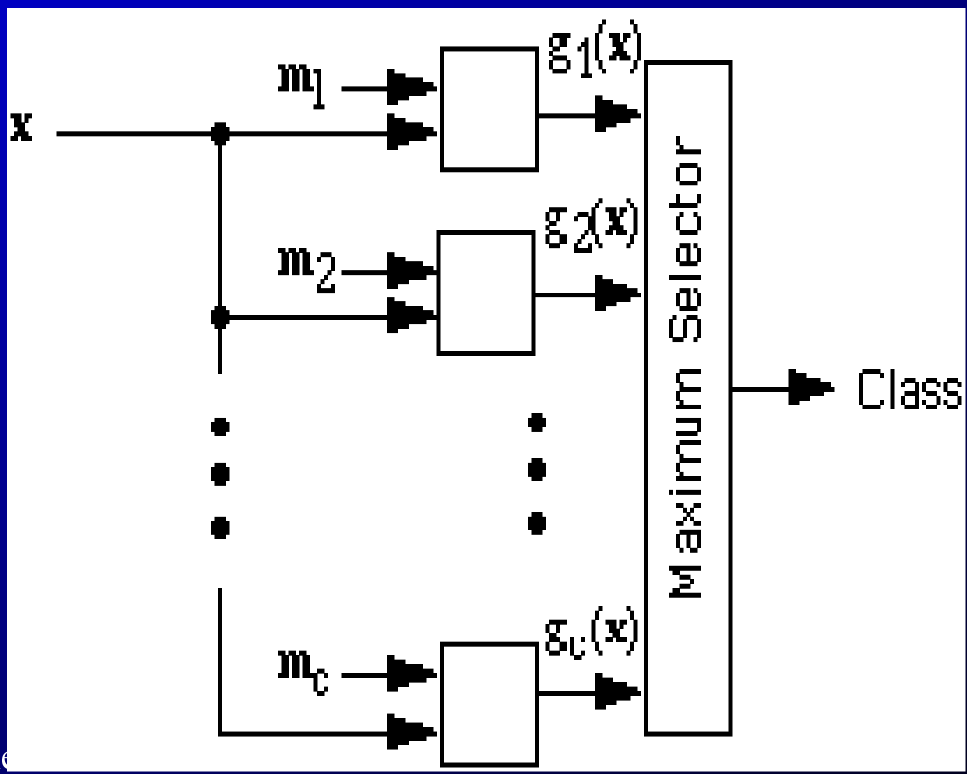
Notice that the **x'x** term is the same for every class, i.e., for every k. To find the template **m**k that minimizes || **x** - **m**k ||, it is sufficient to find the **m**k that maximizes the bracketed expression, **m**k' **x** - 0.5 **m**k' **m**k.

Linear Discriminants (2)

Let us define the **linear discriminant function** g(**x**) by

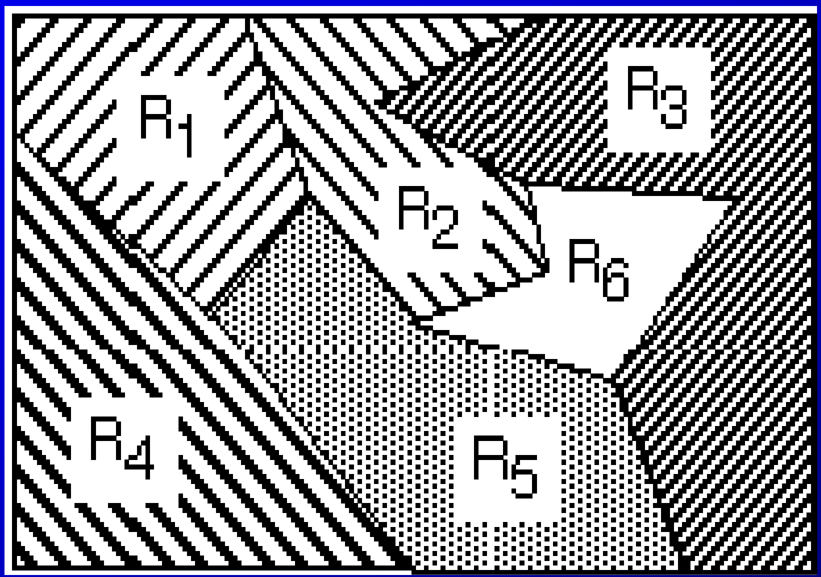
$$g(x) = \mathbf{m}'\,\mathbf{x} - 0.5\ \|\ \mathbf{m}\ \|_2\ .$$

Then we can say that a minimum-Euclidean-distance classifier classifies an input feature vector **x** by computing *c* linear discriminate functions g1(**x**), g2(**x**), ... , gc(**x**) and assigning **x** to the class corresponding to the maximum discriminant function. We can also think of the linear discriminant functions as measuring the correlation between **x** and **m**k, with the addition of a correction for the "template energy" represented by $\|\ \mathbf{m}k\ \|_2$. With this correction included, a minimum-Euclidean-distance classifier is equivalent to a maximum-correlation classifier.



Decision Boundaries

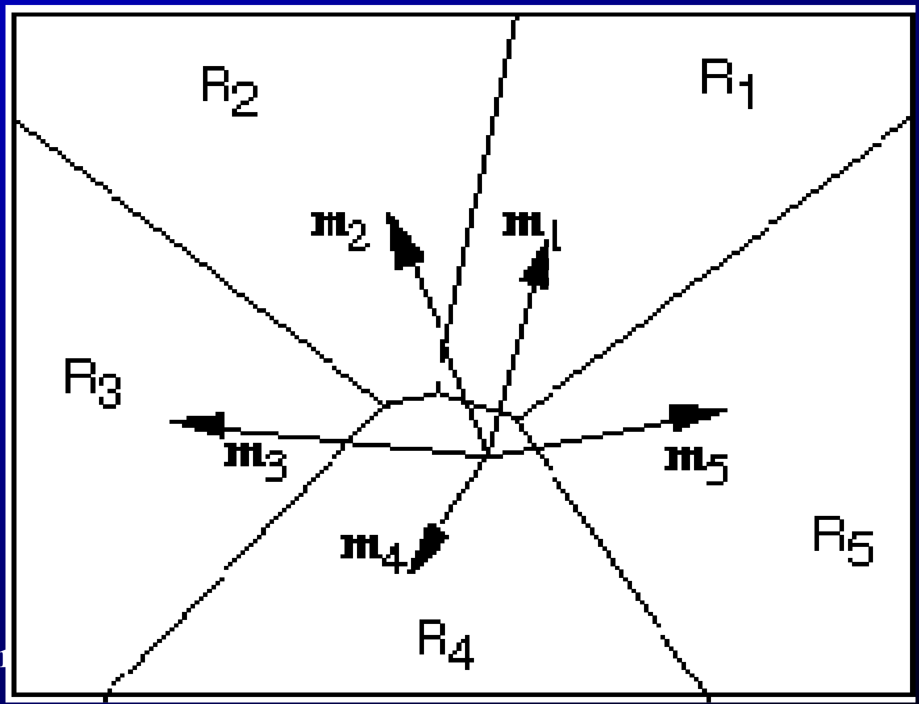
In general, a pattern classifier carves up (or tesselates or partitions) the feature space into volumes called **decision regions**. All feature vectors in a decision region are assigned to the same category. The decision regions are often simply connected, but they can be multiply connected as well, consisting of two or more non-touching regions.



The decision regions are separated by surfaces called the **decision boundaries**. These separating surfaces represent points where there are **ties** between two or more categories.

Decision Boundaries (2)

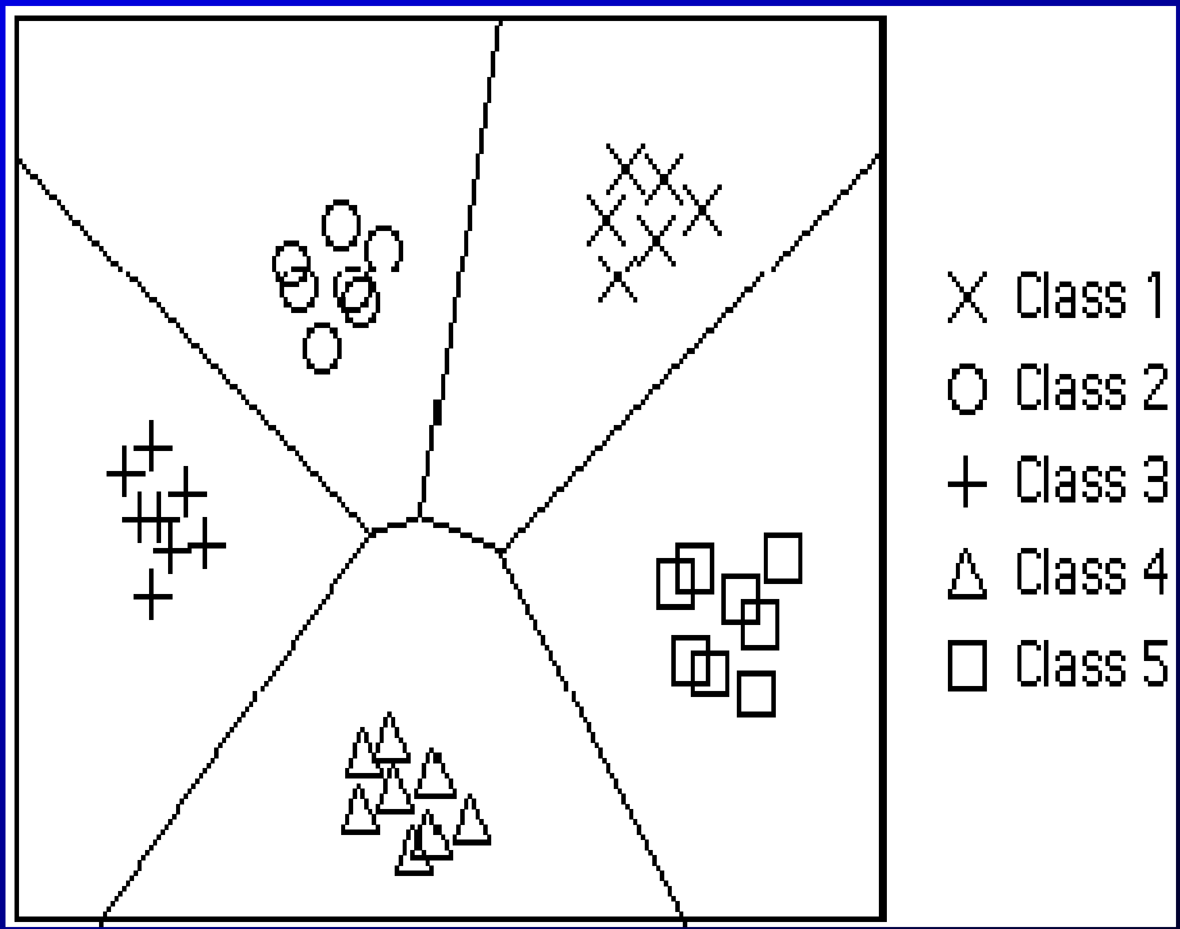
For a minimum-distance classifier, the decision boundaries are the points that are equally distant from two or more of the templates. With a Euclidean metric, the decision boundary between Region i and Region j is on the line or plane that is the perpendicular bisector of the line from **mi** to **mj**. Analytically, these linear boundaries are a consequence of the fact that the discriminant functions are linear. (With the Mahalanobis metric, the decision boundaries are quadratic surfaces, such as ellipsoids, paraboloids or hyperboloids.)



Nearest-template decision boundaries

Decision Boundaries (3)

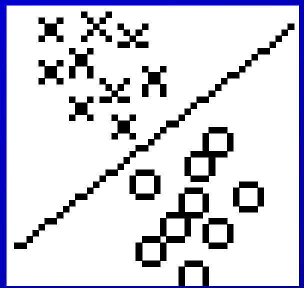
How well the classifier works depends upon how closely the input patterns to be classified resemble the templates. In the example sketched below, the correspondence is very close, and one can anticipate excellent performance. However, things are not always this good in practice, and one should understand the limitations of simple classifiers.

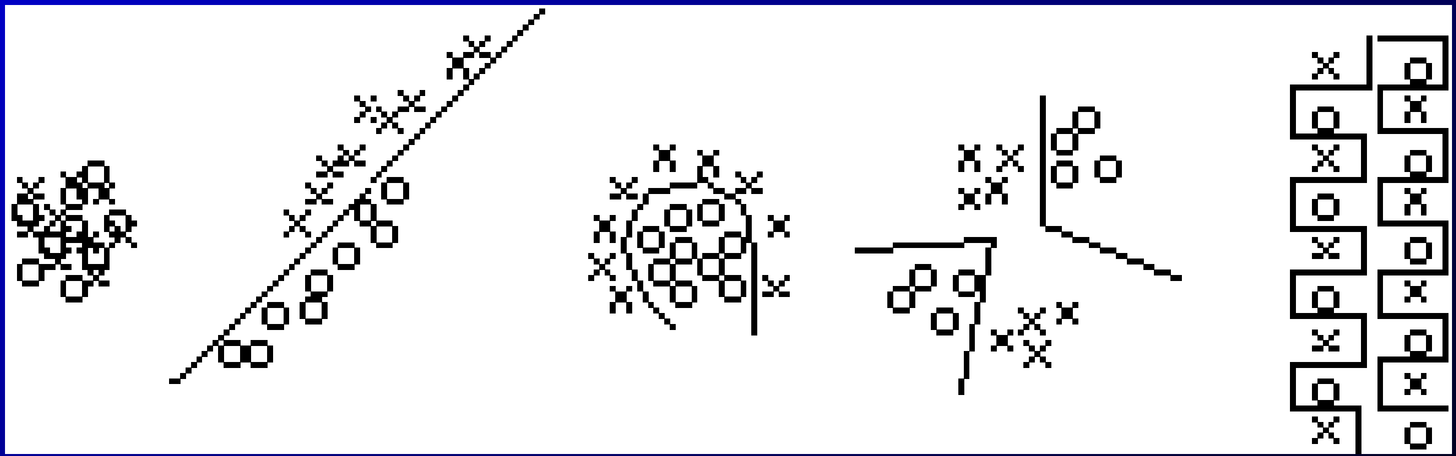


Simple Classifiers: Limitations

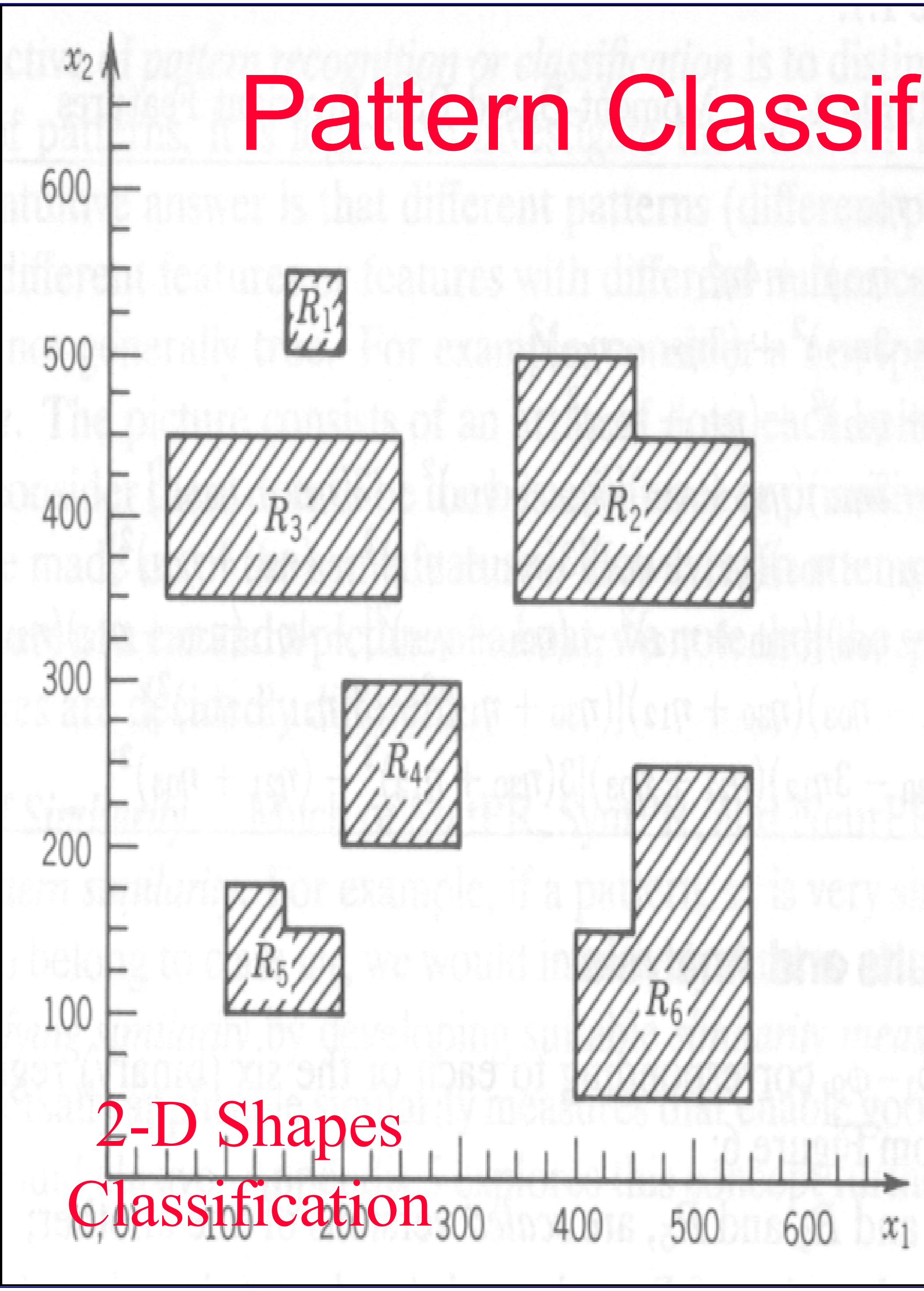
If a simple minimum-distance classifier is satisfactory, there is no reason to use anything more complicated. However, it frequently happens that such a classifier makes too many errors. There are several possible reasons for this:

- The features may be inadequate to distinguish the different classes
- The features may be highly correlated
- The decision boundary may have to be curved
- There may be distinct subclasses in the data
- The feature space may simply be too complex

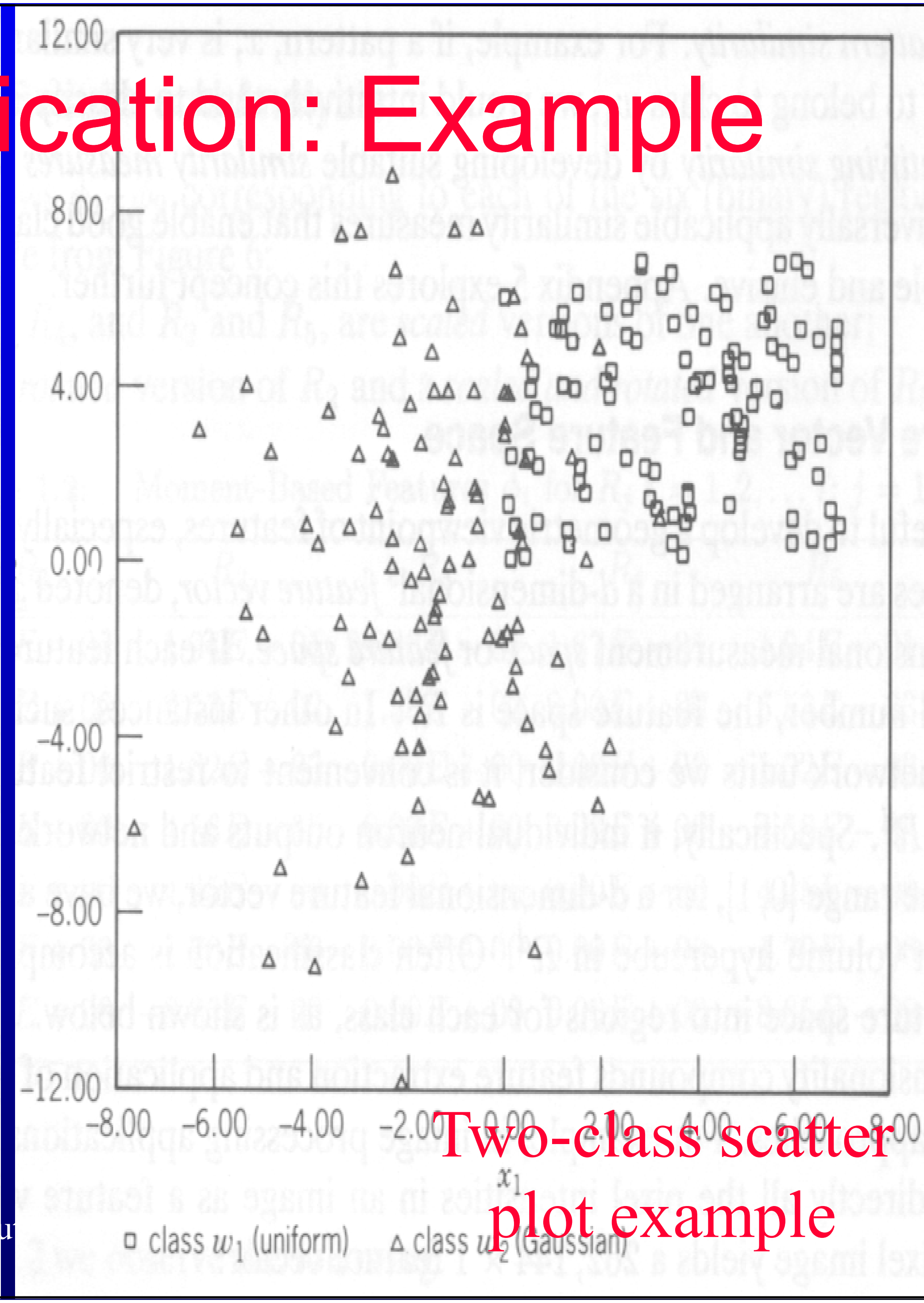




Pattern Classification: Example



2-D Shapes Classification



Two-class scatter plot example

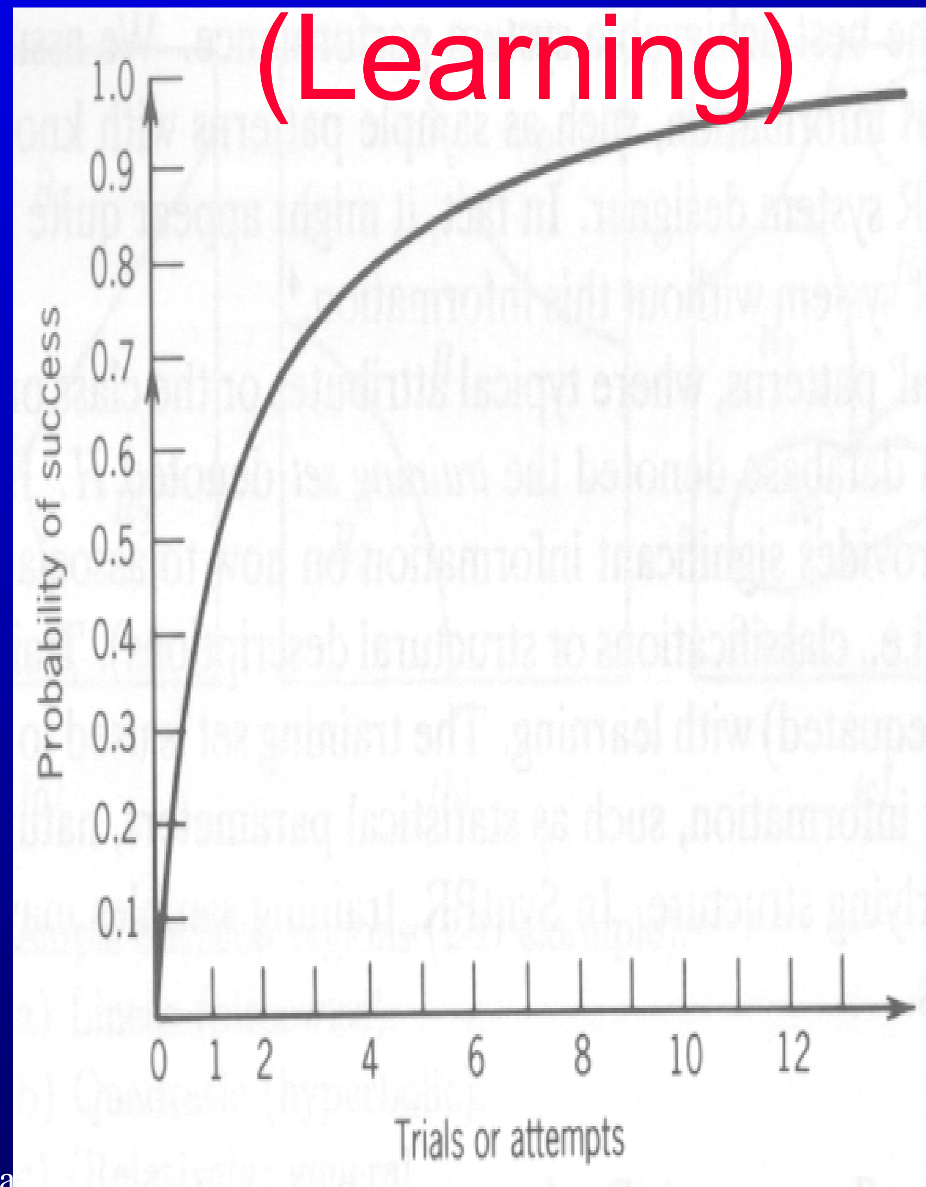
Training (Learning)

- *Usually some data that have already been classified or described are assumed available in order to train the system (the so-called training set).*
- A set of 'typical' patterns, where typical attributes or the class or structure of each is known, forms a database denoted the training set (H).
- Training set provides significant information on how to associate input data with output decisions (i.e., classification or structural descriptions)
- Training set is used to enable the system to 'learn' relevant information, such as statistical parameters, natural groupings, key features, or underlying structure.

Learning Curves (Improved 'Performance')

Training (Learning)

- The difference between desired and actual system output may be used as the performance measure.
- This generic learning concept is related to the error-correction-based PR techniques we employ in developing linear discriminate functions in Statistical PR and the generalized delta rule in Neural PR.
- Both of these techniques are typical of gradient-descent techniques, where the system is modified following each experiment or iteration.
- This may lead to the typical 'learning curve'.



Questions?

1. What is “Pattern”? According to its definition, could you list some patterns in your daily life?
2. “Pattern Recognition (PR)” is defined in this lecture. Could you give a PR samples (such as biometrics) and tell which features you use? .
3. Why PR system needs a training process? If the training set is too small, what happened? Why The difference between desired and actual system output may be used as the performance measure?
4. Try to design a classifier using the selected technology and tell why?

So much for today!

Thank you !!!