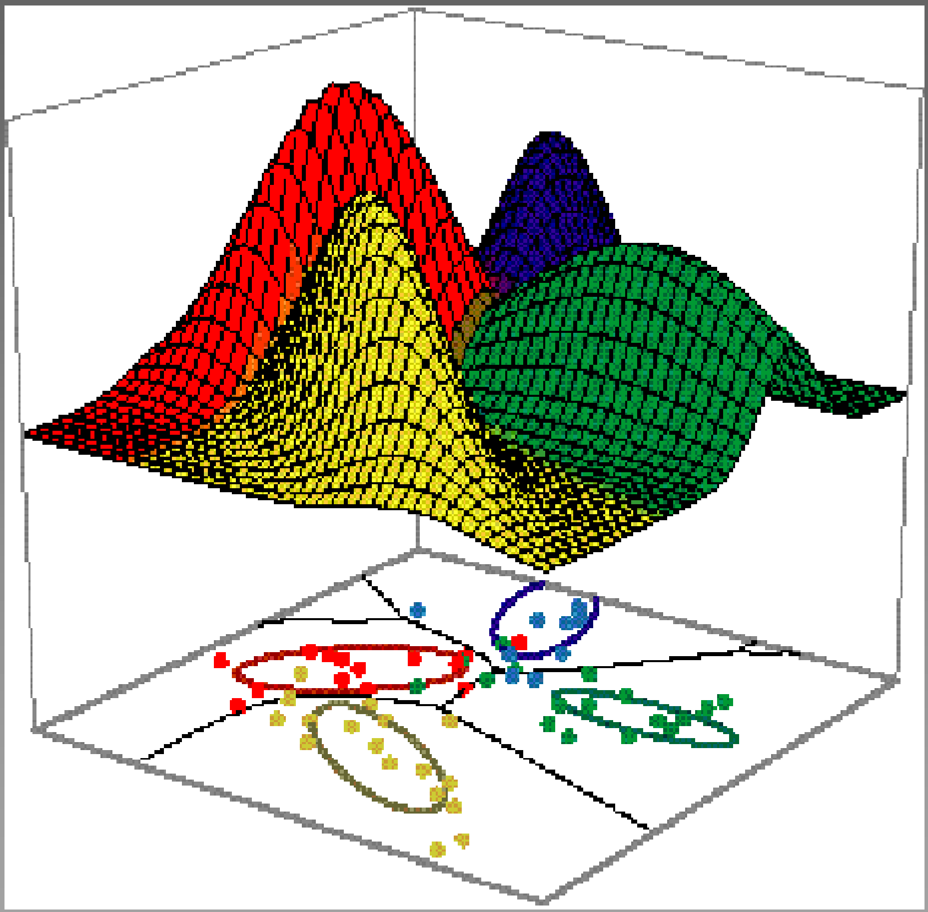


Pattern Reco

Pattern Classificatio

by R. O. Duda, P. E. Hart a

John Wiley & Son

Chapter 1: Introduction to Pattern Recognition (Sections 1.1 - 1.6)

- Machine Perception
- An Example
- Terminology
- Pattern Recognition Systems
- The Design Cycle
- Learning and Adaptation
- Conclusion

MACHINE PERCEPTION

- **Build a machine that can recognize patterns**:

- Optical Character Recognition
  - Sorting letters by postal code
  - Reconstructing text from printed materials (reading machines for blind people)
  - Data Mining (text)
- Analysis and identification of human patterns
  - Speech and voice recognition.
  - Finger prints and DNA mapping.
  - Human motion, emotion and age detection
- Banking and insurance
  - Credit card applicants classified by income, credit, mortgage amount, # of dependents, etc.
  - Car insurance (pattern like mark of car, # of accidents, age, driving habits, location…)
- Diagnosis systems
  - Medical diagnosis (disease vs. Symptoms classification, X-Ray, MRI, tests analysis, etc.)
  - Diagnosis of automotive malfunctioning.
  - Tumor and cyst segmentation and classification.
- Prediction systems
  - Weather forecasting (based on satellite data).
  - Analysis of seismic patterns.
  - Animals behavior analysis.
- Security and military applications
  - Analysis of network traffic patterns.
  - Recognizing missiles, military vehicles using aerial photographs
  - Radar signal classification

AN EXAMPLE

- “Sorting incoming Fish on a conveyor according to species using optical sensing”

Species → Sea bass

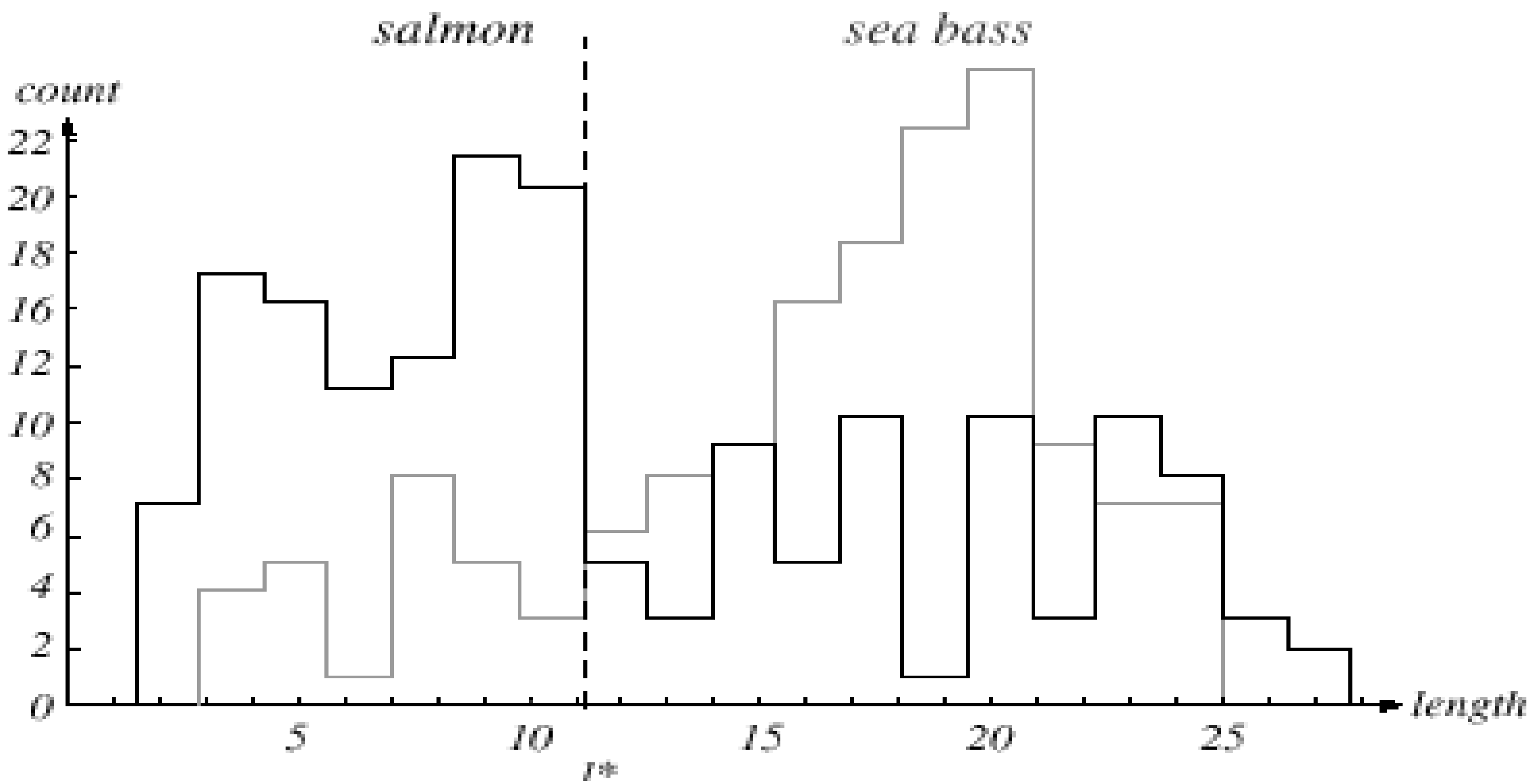
Species → Salmon

- Problem Analysis
  - Set up a camera and take some sample images to extract features
    - Length
    - Lightness
    - Width
    - Number and shape of fins
    - Position of the mouth, etc…
    - This is the set of all suggested features to explore for use in our classifier!

- Preprocessing
  - Use a segmentation operation to isolate fishes from one another and from the background
- Information from a single fish is sent to a feature extractor whose purpose is to reduce the data by measuring certain features
- The features are passed to a classifier

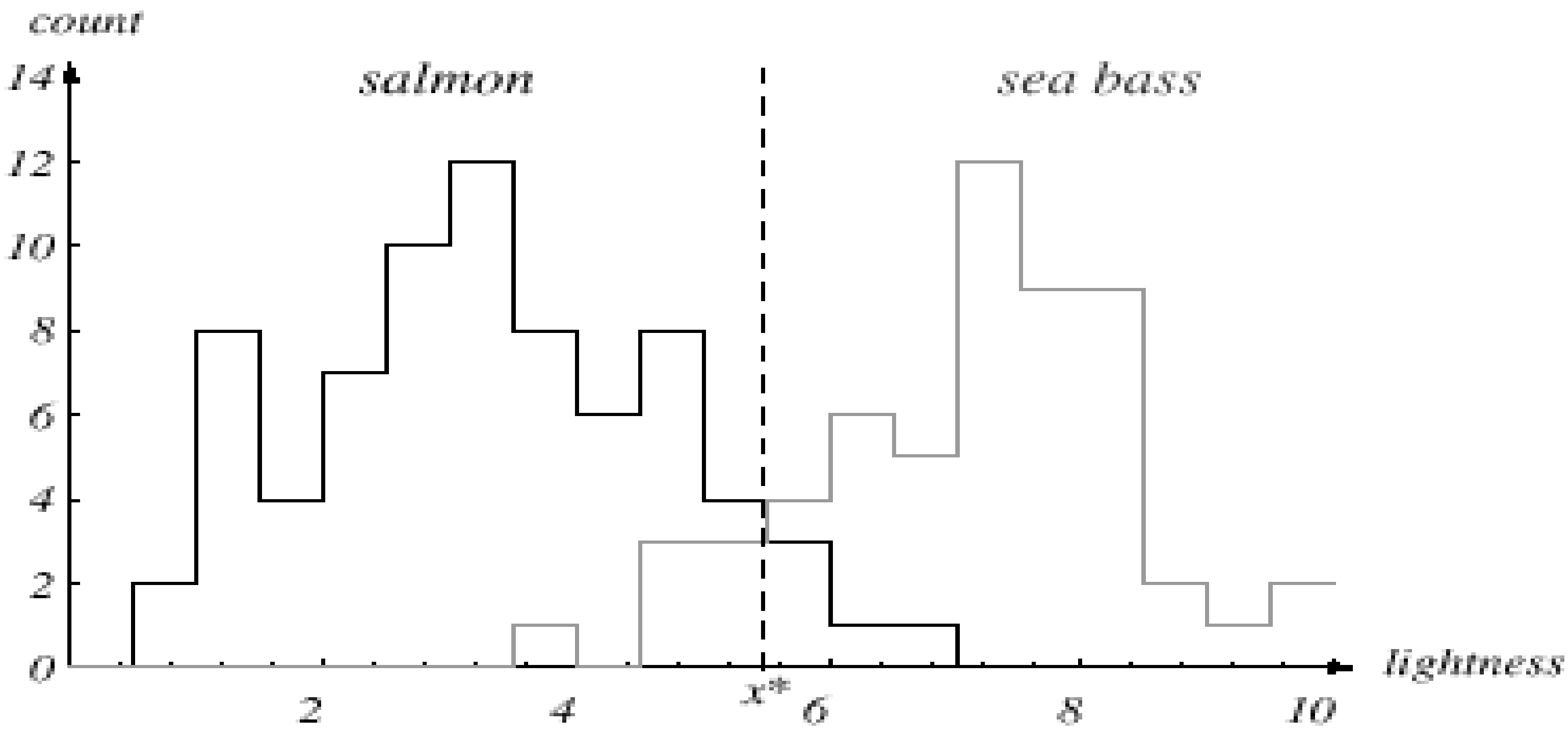

Preprocessing
Feature extraction
Classification
"salmon"
"sea bass"

- Classification
  - Select the length of the fish as a possible feature for discrimination


salmon
sea bass
count
22
20
18
16
12
10
8
6
4
2
0
5
10
15
20
25
l*
length

The length is a poor feature alone!

Select the lightness as a possible feature.


count
salmon
sea bass
14
12
10
8
6
4
2
0
lightness
2
4
x*
6
8
10

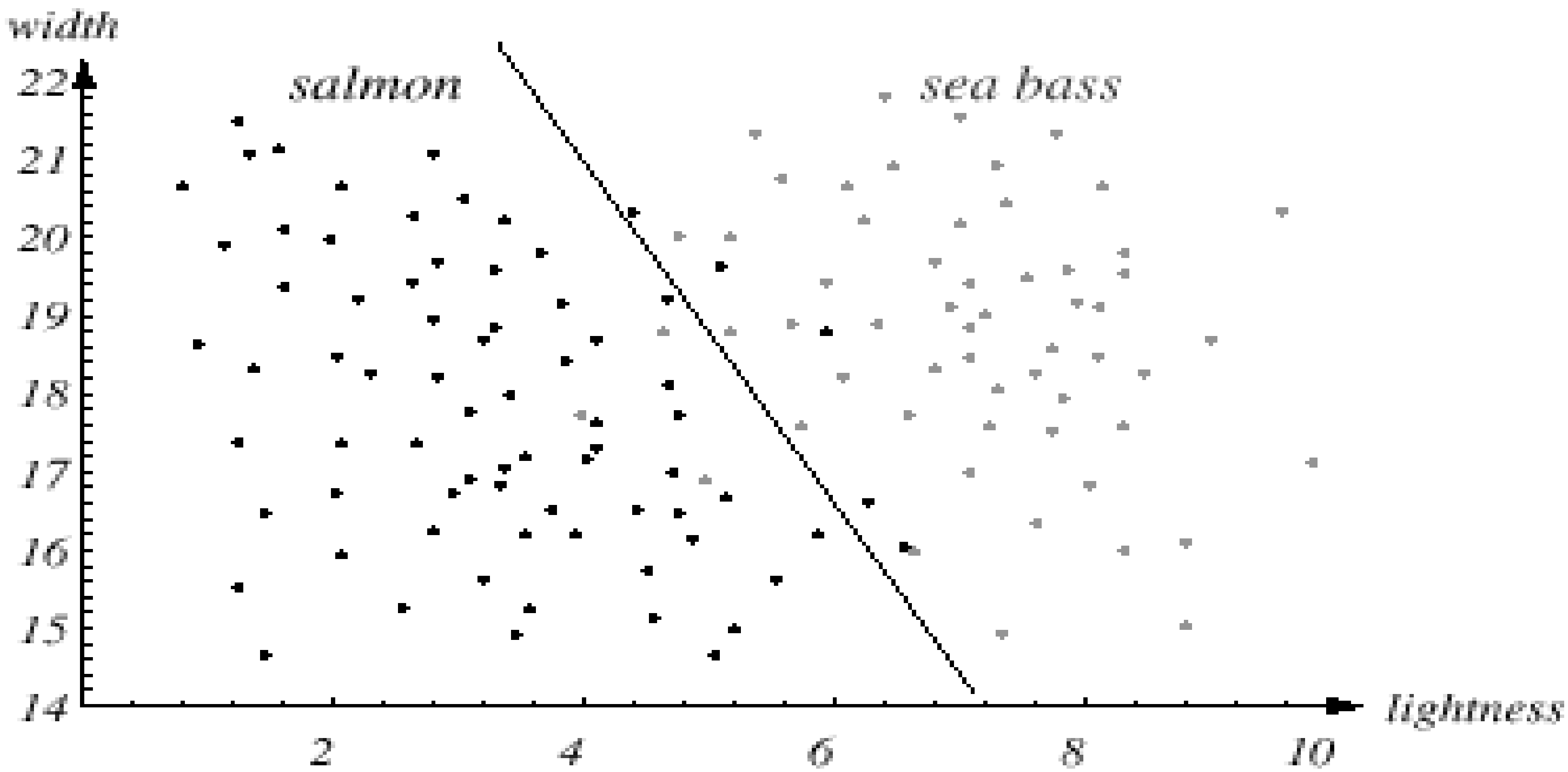
- Threshold decision boundary and cost relationship
  - Move our decision boundary toward smaller values of lightness in order to minimize the cost (reduce the number of sea bass that are classified salmon!)

Task of decision theory

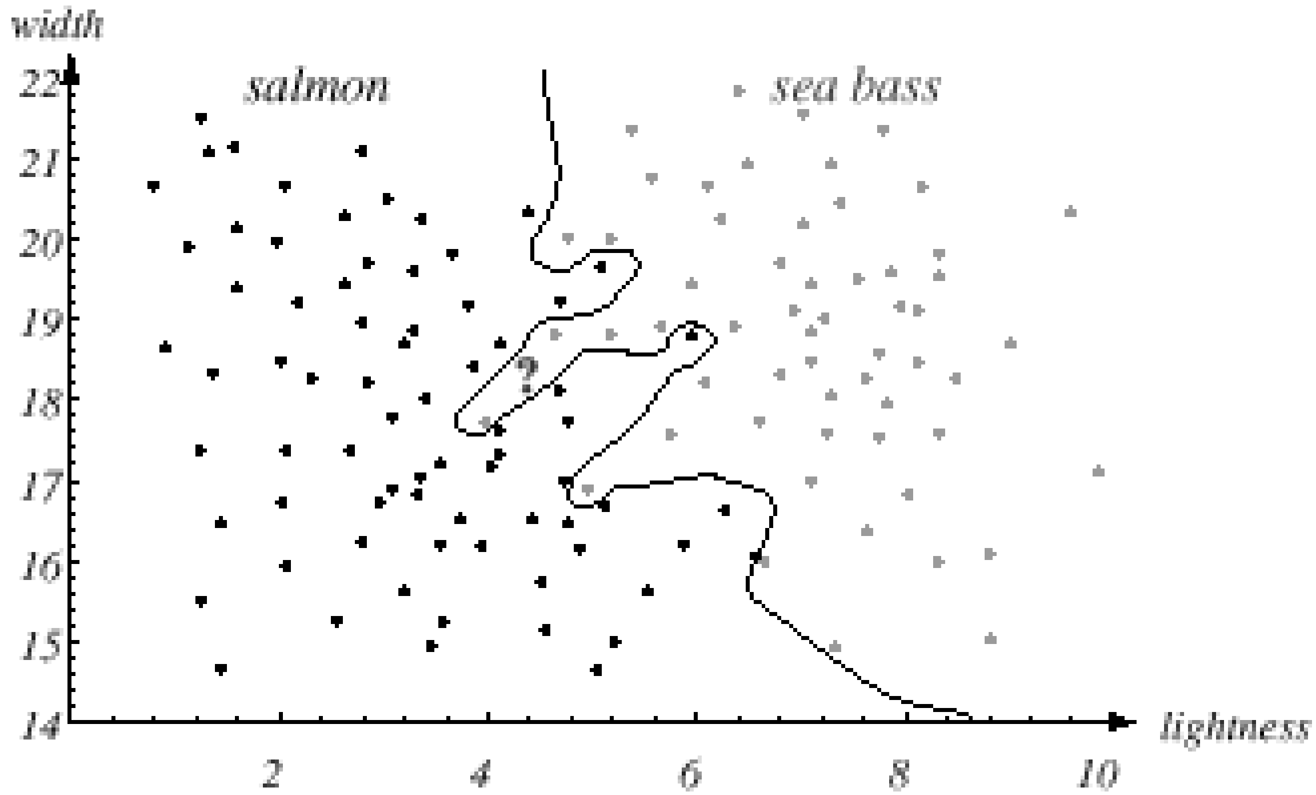
- Adopt the lightness and add the width of the fish

Fish $\Rightarrow$ $x^T = [x_1, x_2]$

Lightness **Width**

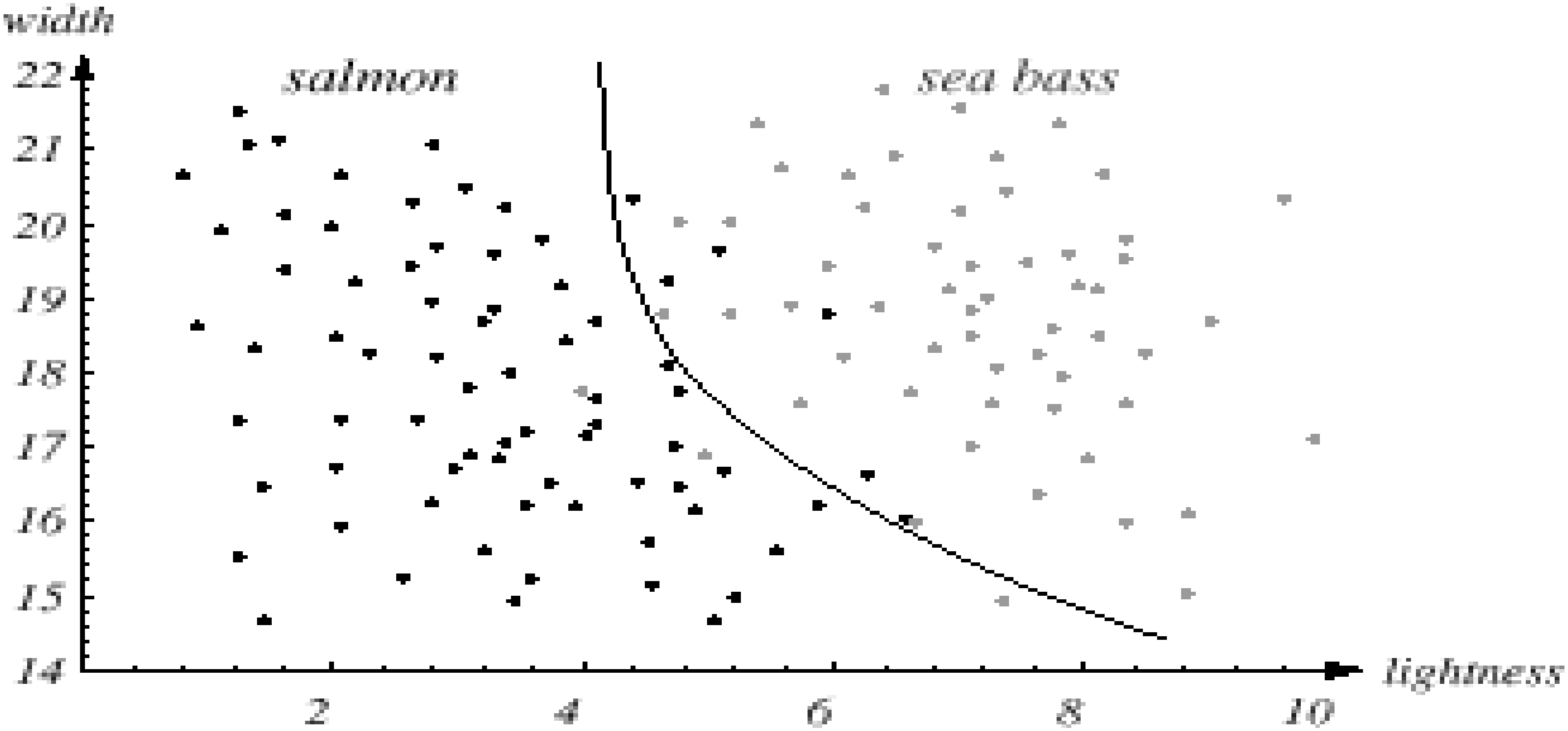

width
22
21
20
19
18
17
16
15
14
salmon
sea bass
2
4
6
8
10
lightness

- We might add other features that are not correlated with the ones we already have. A precaution should be taken not to reduce the performance by adding such “noisy features”
- Ideally, the best decision boundary should be the one which provides an optimal performance such as in the following figure:


width
salmon
sea bass
lightness
22
21
20
19
18
17
16
15
14
2
4
6
8
10
?

- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel input

Issue of generalization!


width
22
21
20
19
18
17
16
15
14
salmon
sea bass
2
4
6
8
10
lightness

TERMINOLOGY

- Pattern: object, process or event consisting of both deterministic and stochastic components; a record of dynamic occurrences influenced by both deterministic and stochastic factors.
- Pattern class: set of patterns sharing a set of common attributes (or features) and usually originating from the same source (associated with the generalization or abstraction of patterns).
- Feature: relevant (intrinsic) trait or characteristic that makes a pattern apart from another; data extractable through measurement and/ or processing.
- Examples:
  - **Patterns:** textures, crystals, biology patterns, constellations, weather pattern, speech waveform, etc.
  - **Features:** color, age, weight, aspect ratio, etc.
- Classification: assigning patterns into classes based on features.
- Recognition: ability to classify patterns.
- Errors: Confusion patterns.
- Noise: distortions associated with pattern processing (errors in feature extraction) and/ or training samples that impact the classification (i. e. recognition) abilities of the system.

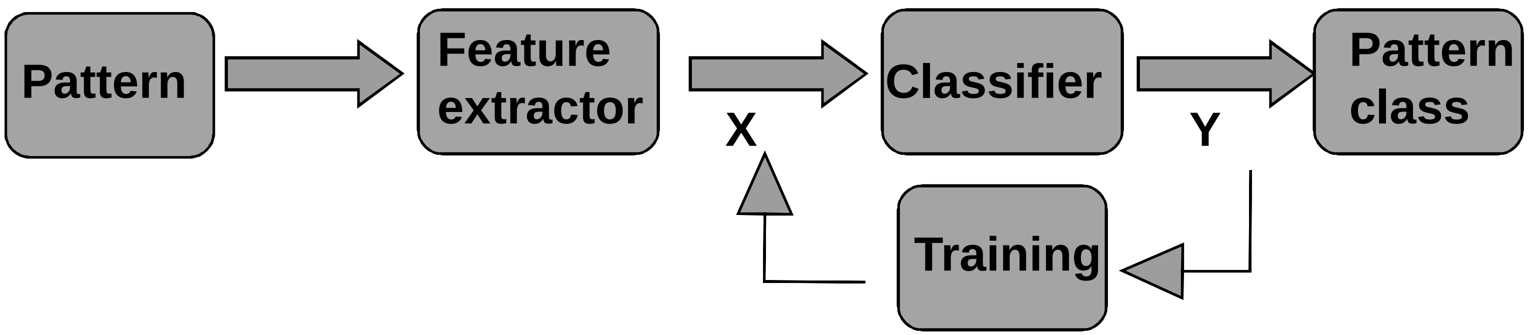
General Problem

- Given an input set of features (or input vector) extracted from a sample pattern:

$$X = \{x\ 1\ ,\ x\ 2\ ,\ \ldots,\ x\ n\ \}$$

- find the associated set of characteristics (output vector) matching the input according to a pre- defined criteria:

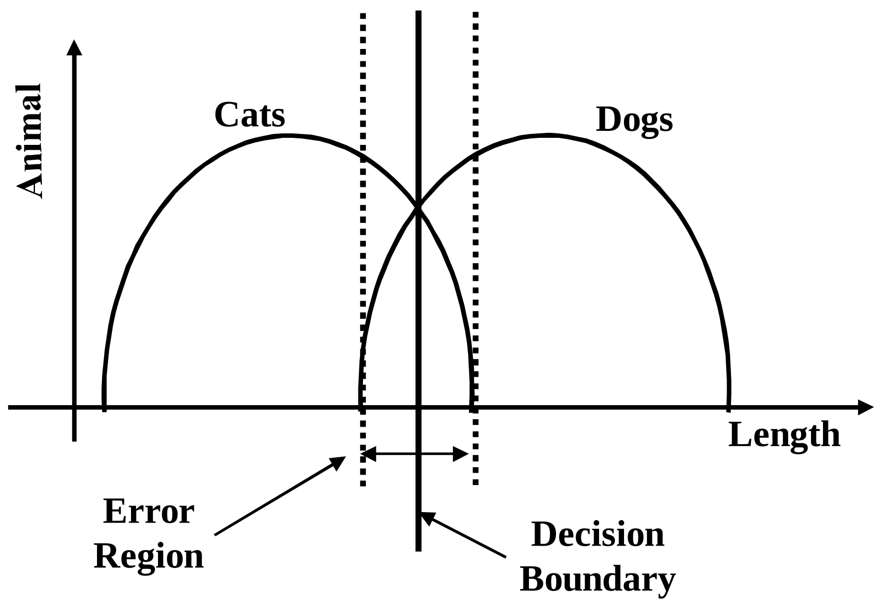
$$Y = \{y\ 1\ ,\ y\ 2\ ,\ \ldots,\ y\ m\ \}$$



- It is assumed that a set of training samples was previously used to "teach" the recognition system how to link input and output vectors.

Decision Boundary

- How to distinguish a cat from a dog? Suppose we choose to make the classification based on the length of the animal (from tail to snout). Then, the length is our feature for classification.

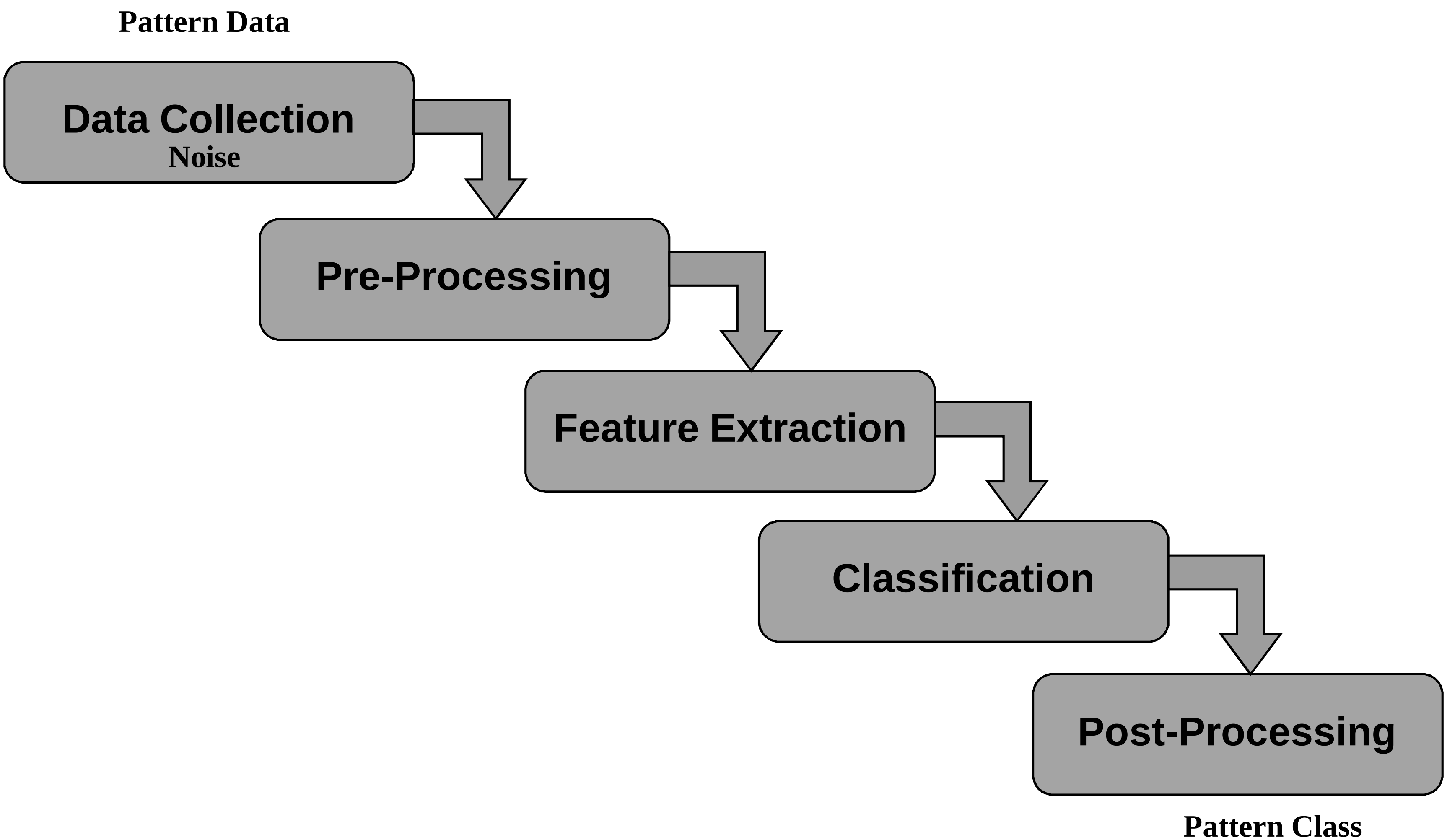


- Boundary can be linear or non-linear

Classifiers

- A classifier partitions the feature space into class-labeled decision regions. The border of each region is a decision boundary.
- Ideally, a unique class assignment is achieved if, and only if, the decision regions cover the entire feature space and they are disjoint (i. e. the decision boundaries do not create error or confusion areas).
- Classification strategy:
  - Extract the feature vector of the pattern.
  - Compare the feature vector with the characteristics of existing decision regions (i. e. with features of each pattern class).
  - Assign pattern to the class (associated with decision region) matching the feature vector extracted.

PATTERN RECOGNITION SYSTEMS



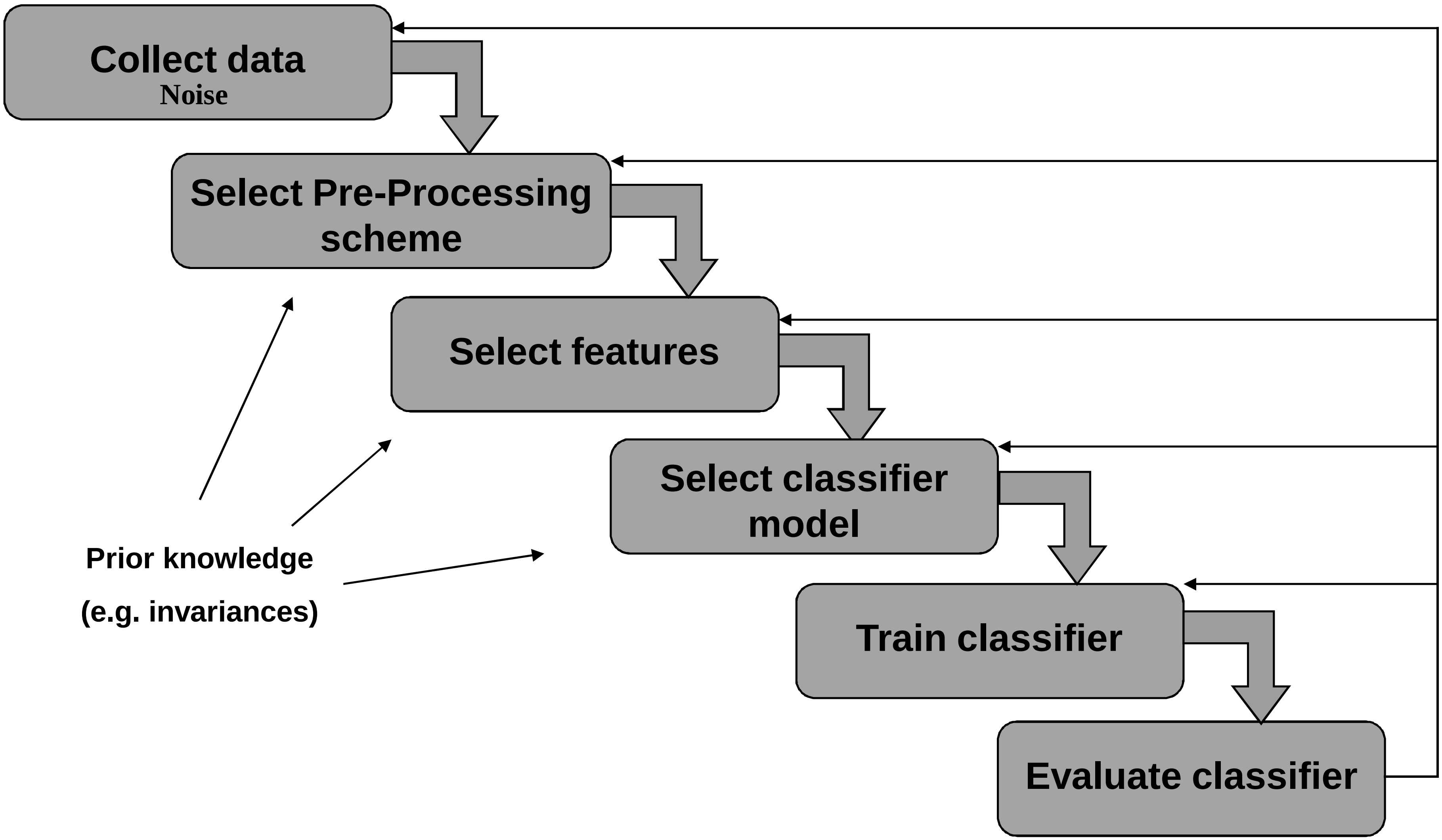
Pattern Recognition Systems

- Data Collection
  - Use of a transducer (camera, microphone, etc.)
  - PR system depends on the bandwidth, the resolution, sensitivity distortion of the transducer

- Pre-Processing
  - Data should be well separated and should not overlap
  - Noise filtering
  - Segmentation and grouping
  - Size normalization
  - Thinning and skeletonization

- Feature Extraction
  - Discriminative features
  - Invariant features with respect to translation, rotation and scale
  - Uncorrelated features
- Classification
  - Use a feature vector provided by a feature extractor to assign the object to a category
  - Adjust for missing feature
  - Applying recognition scheme
  - Match pattern to class
- Post-Processing
  - Exploit context input dependent information other than from the target pattern itself to improve performance
  - Apply correction techniques

THE DESIGN CYCLE



- Data Collection
  - How do we know when we have collected an adequately large and representative set of examples for training and testing the system?

- Pre-processing scheme selection
  - Choosing an optimum noise filtering algorithm, well separating the patterns using segmentation and grouping

- Feature selection
  - Depends on the characteristics of the problem domain. Simple to extract, invariant to irrelevant transformation and insensitive to noise.

- Classifier model selection

  - Unsatisfied with the performance of our fish classifier and want to jump to another class of model.

- Training

    - Use data to determine the classifier. Many different procedures for training classifiers and choosing models

- Evaluation
    - Measure the error rate (or performance and switch from one set of features to another one).

- Computational complexity
  - What is the trade-off between computational ease and performance?
  - How an algorithm scales as a function of the number of features, patterns or categories?

Learning

- **Problem:** we fit a model based on past experience (past examples seen), but we are ultimately interested in learning the mapping that performs well on the entire population of examples.
- **Training:** data is used to adjust (iteratively) the parameters of the model until it fits; training is considered accomplished when overall error falls below a pre-define threshold.
- **Training error:**

$$\Sigma\, (y_i - f(x_i))^2$$

- **Generalized error** (over the whole population):

$$E_{x,y}\, (y - f(x))^2$$

- *Minimizing the training error does not imply a low generalized error*

LEARNING AND ADAPTATION

- Supervised learning
  - the classifier system is “taught” to associate patterns with their respective classification labels (provided along with each training pattern).
- Unsupervised learning
  - the system is directed (based on some pre- defined hypothesis) to form clusters (or “natural groupings”) of the training patterns (that are not labeled).
- Reinforced learning
  - training is assisted by a “critic” that provides a binary feedback by informing the system whether the classification associated with a given training sample is either right or wrong.

CONCLUSION

- We are facing the complexity and magnitude of the sub-problems of Pattern Recognition
- Many of these sub-problems can indeed be solved
- Many fascinating unsolved problems still remain

Next

- Traditional Pattern Recognition Techniques:
  - Parametric Methods: *Bayesian*
    - Each class is described along pre-defined parameters and it is identified by the parametric distribution it exhibits.
    - Statistical distribution of parameters is learned through training.
  - Non-Parametric Methods: *Nearest-Neighbor, Parzen Windows*
    - Pattern classes are not associated with parametric distributions, but with density functions and/or decision regions within feature space.
    - A crude classification model (usually based on intuition) is iteratively refined through training.