CSCI 5521: Pattern Recognition

Prof. Paul Schrater

Business

- Course web page:

  http://gandalf.psych.umn.edu/~schrater/schrater_lab/courses/PattRecog07/PattRecog.html

Syllabus

- Reading materials:
  - Pattern Classification, 2nd Ed. Duda, Hart, Stork
  - Statistical Pattern Recognition, 2nd Ed. Andrew Webb
  - Elements of Statistical Learning: Data Mining, Inference, Prediction, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Springer, 2001.
  - Gaussian Processes for Machine Learning, Carl Rasmussen and Christopher K.I. Williams, MIT Press, 2006.
  - Papers posted on the web site.
  - Downloads will be password protected.
- Grading
- 60% on the homework assignments
- 40% on the final project.

Syllabus cont'd

- Final Project 12-15 page paper involving:

1) Simulation or experiments. For example, implement a pattern recognition system for a particular application, e.g. digit classification, document clustering, etc.
2) Literature survey (with critical evaluation) on a given topic.
3) Theoretical work (detailed derivations, extensions of existing work, etc)

Important dates:

- **Sept. 28:** Topic selection. One or two pages explaining the project with a list of references.
- **Nov. 7:** Partial report (3 to 5 pages).
- **Dec. 20:** Final report (12 to 15 pages).
- Students may work in groups of 2-4.

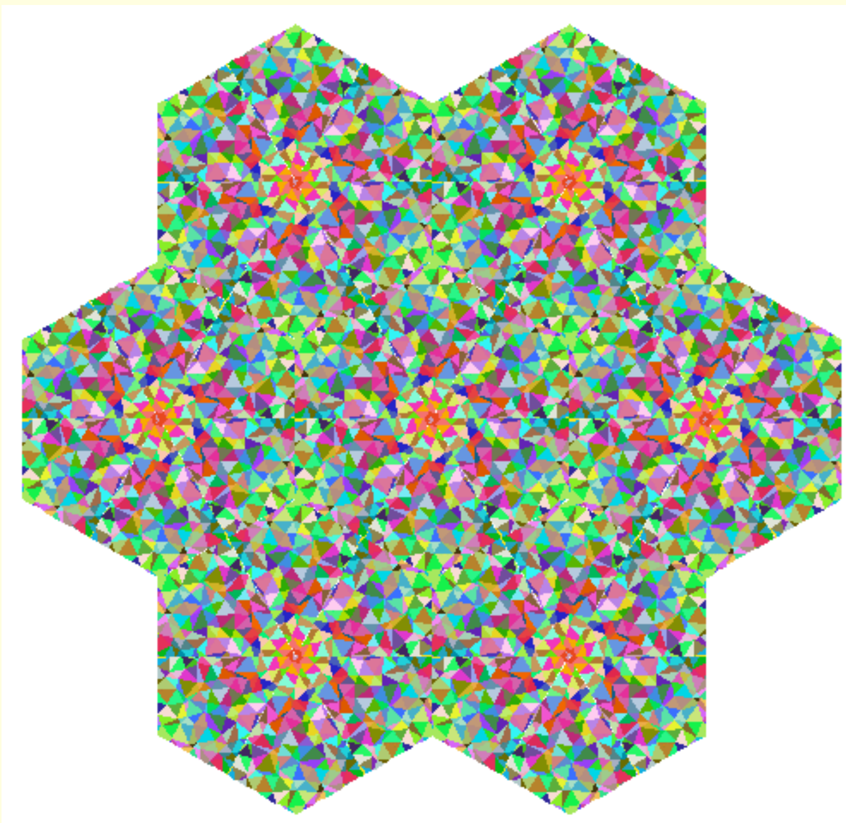
Policies/Procedures

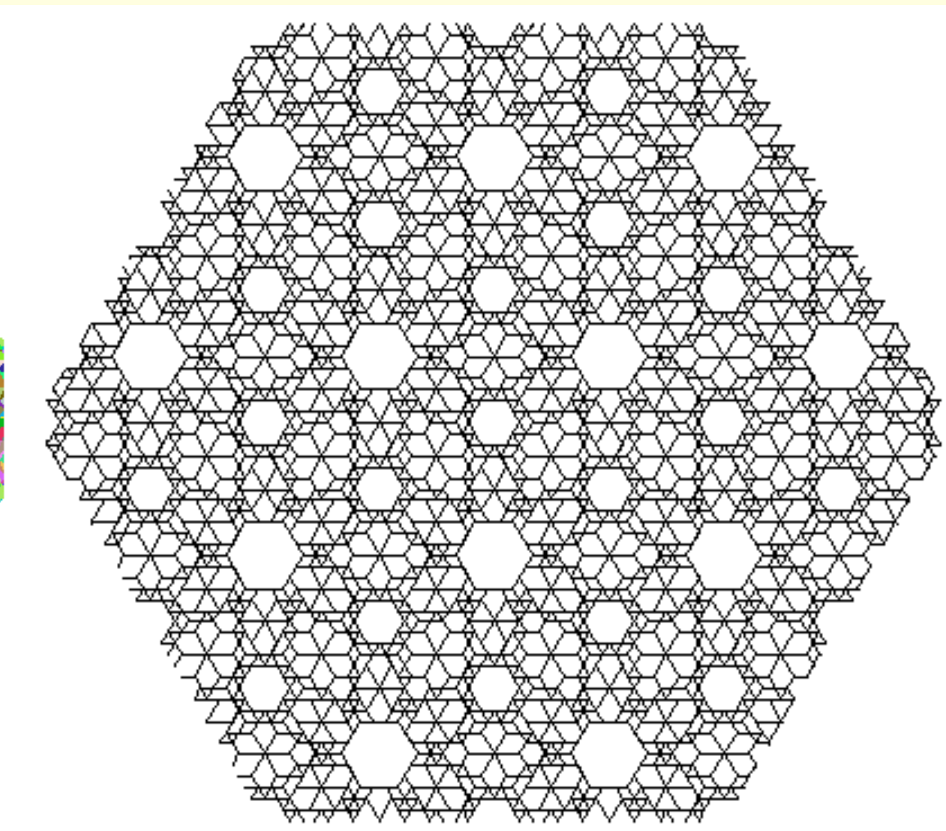
- DO NOT CHEAT.
- Do NOT work in groups for homework (consulting with fellow students permitted, but group submissions are not).
- Electronically submit homework.
- Homework must be submitted by Midnight on the day it is due.

Introduction to Pattern Recognition

- Syllabus
- What are Patterns?
- Pattern Recognition
- An Example
- Pattern Recognition Systems
- The Design Cycle
- Learning and Adaptation
- Conclusion

Examples of Patterns



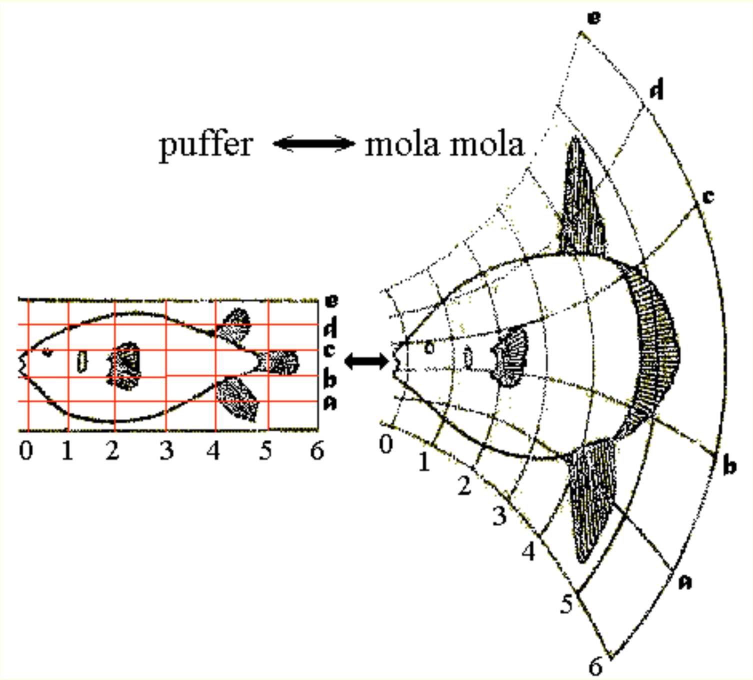


Examples of Patterns

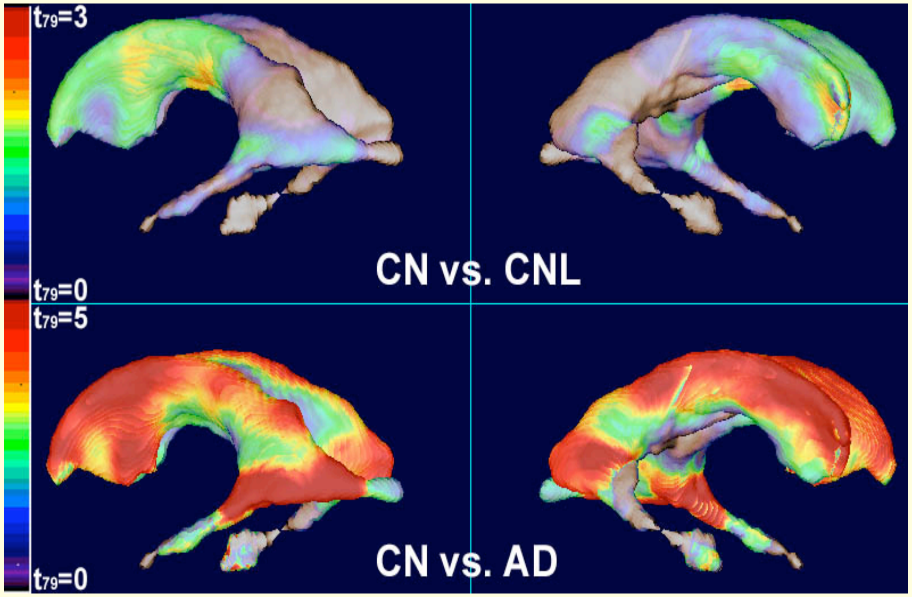
Natural or Not?

How can we describe these patterns?

Shape Patterns



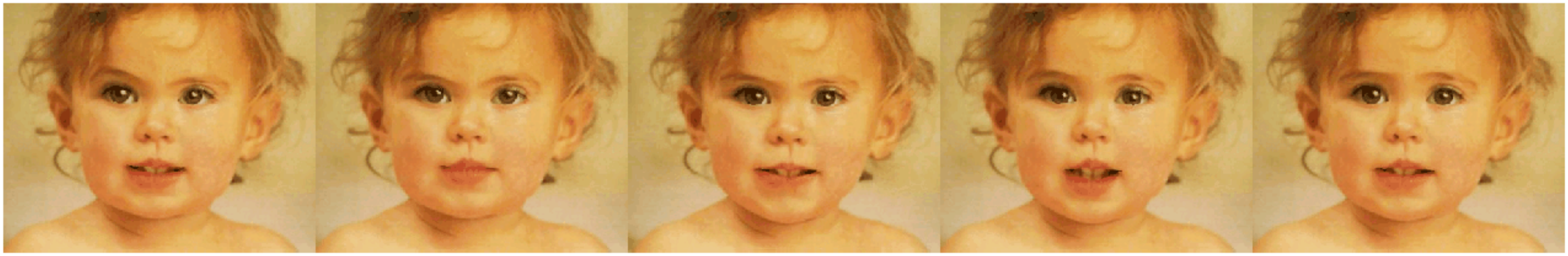
D'arcy Thompson's suggestion of species change through continuous deformation



This figure shows the effects of Alzheimer's Disease on the ventricular expansion rate measured from serial MRI.

Explaining patterns

Voice Puppetry, M. Brand; Siggraph'99



Frames from a voice-driven animation, computed from a single baby picture and an adult model of facial control. Note the changes in upper facial expression. See figures 4.5, 5 and 8 for more examples of predicted mouth shapes.

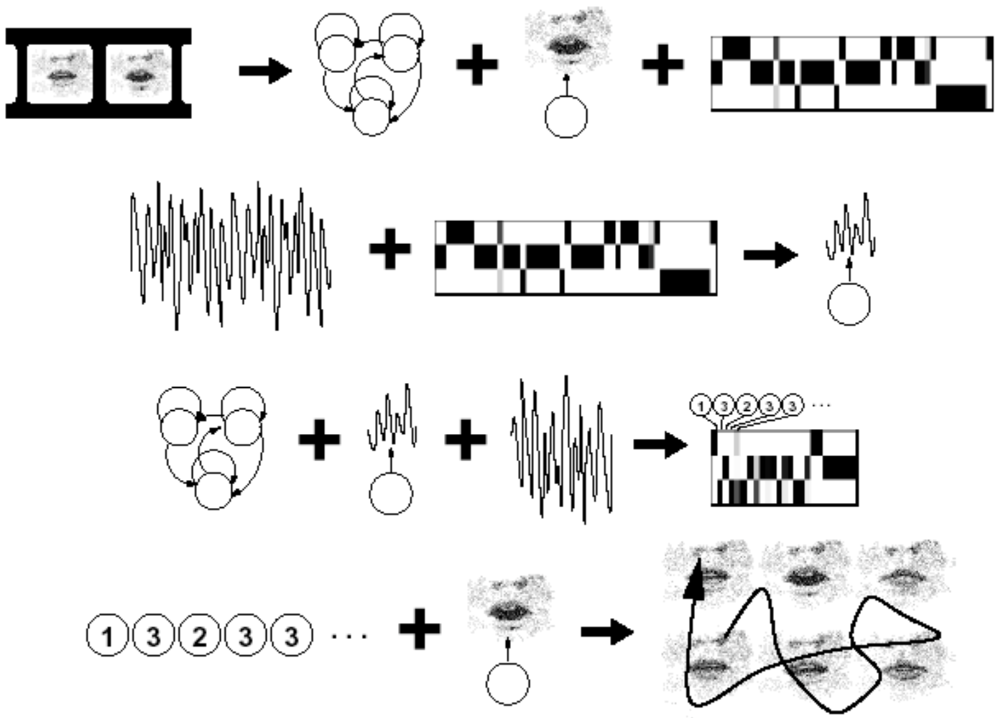


Figure 1: Schematic of the training, remapping, analysis, and synthesis steps of voice puppetry. See §4 overview.

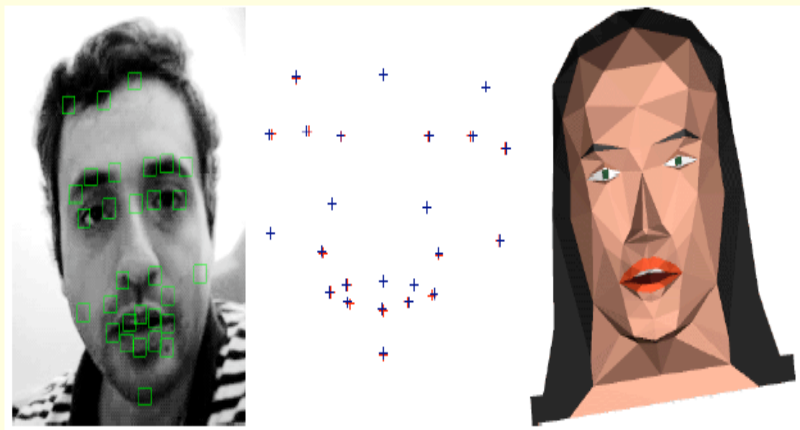
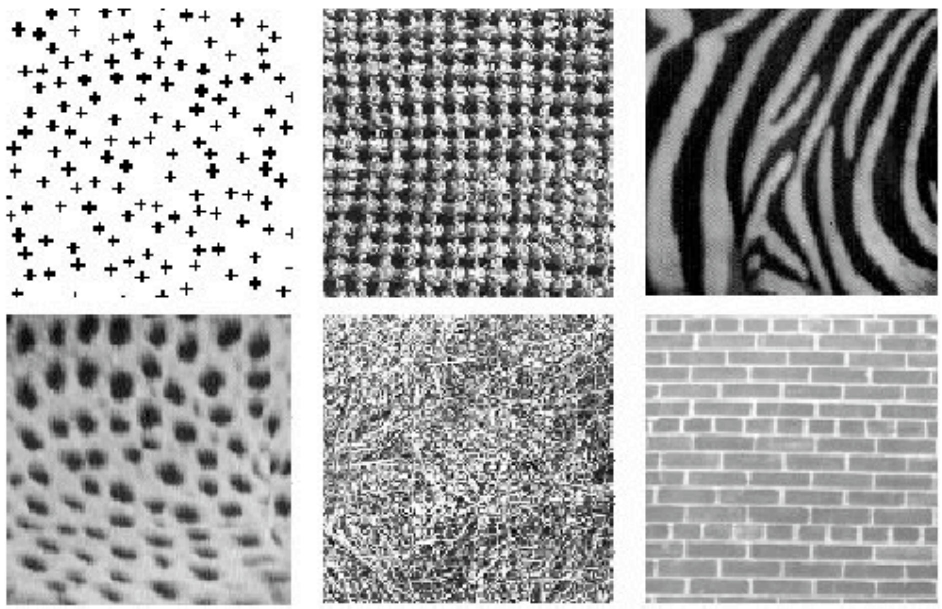


Figure 2: (A) Tracking of facial features for training data. See §4.1. (B) Voice-predicted facial feature locations (blue) superimposed over ground-truth locations (red). See §4.5. (C) A 3D model articulated via vertex motions. See §4.6.

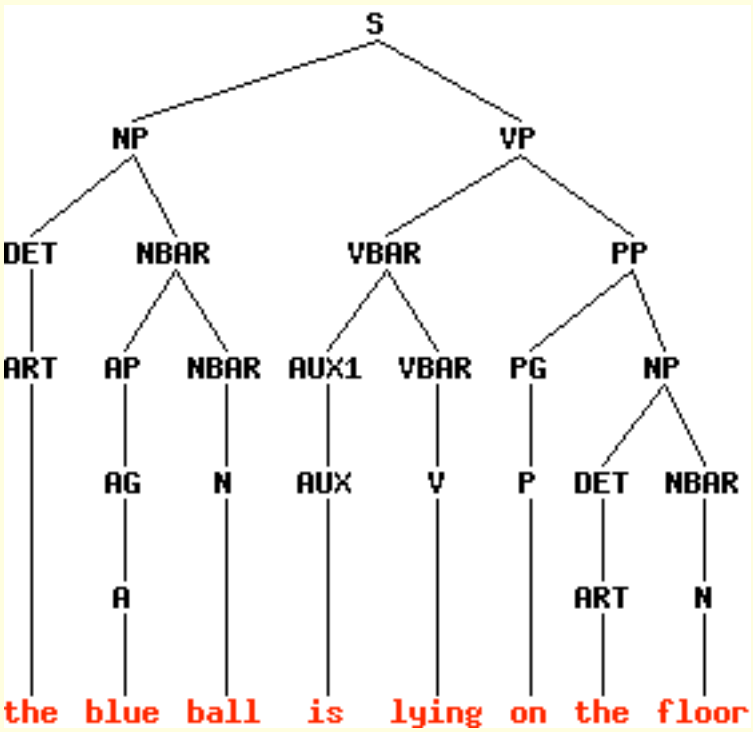
Pattern Examples

A wide variety of texture patterns are generated by various stochastic processes

How are these patterns represented in human brain?



Pattern Examples



Natural Language is a pattern

What is a Pattern?

- A set of instances that:
  - Share some regularities and similarities.
  - Are Repeatable.
  - Are Observable, sometimes partially, using sensors with noise and distortions.
- How do we define “regularity”?
- How do we define “similarity”?
- How do we define “likelihood” for the repetition of a pattern?
- How do we model the sensors?
- What is not a pattern?

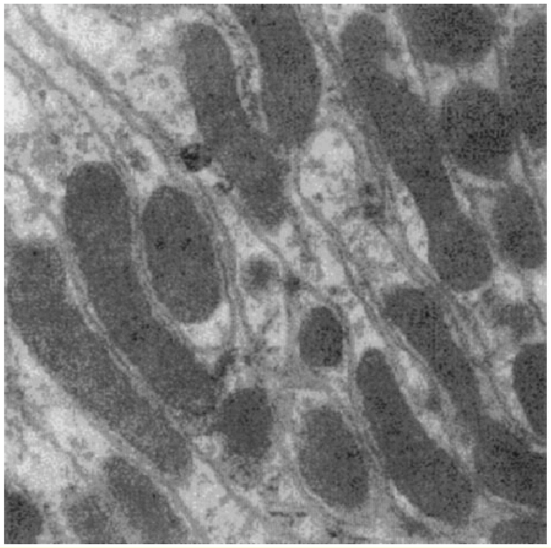
Defining Patterns

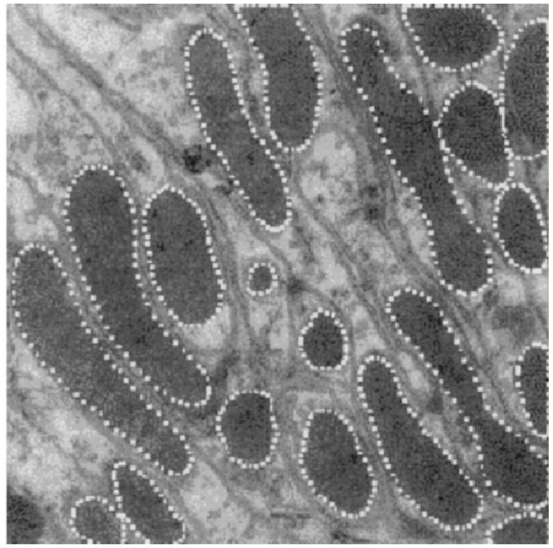
In a mathematical notation, Ulf Grenander (1976-1995) propose to define patterns as follows:

Regularity R=<G, S, ρ, Σ>

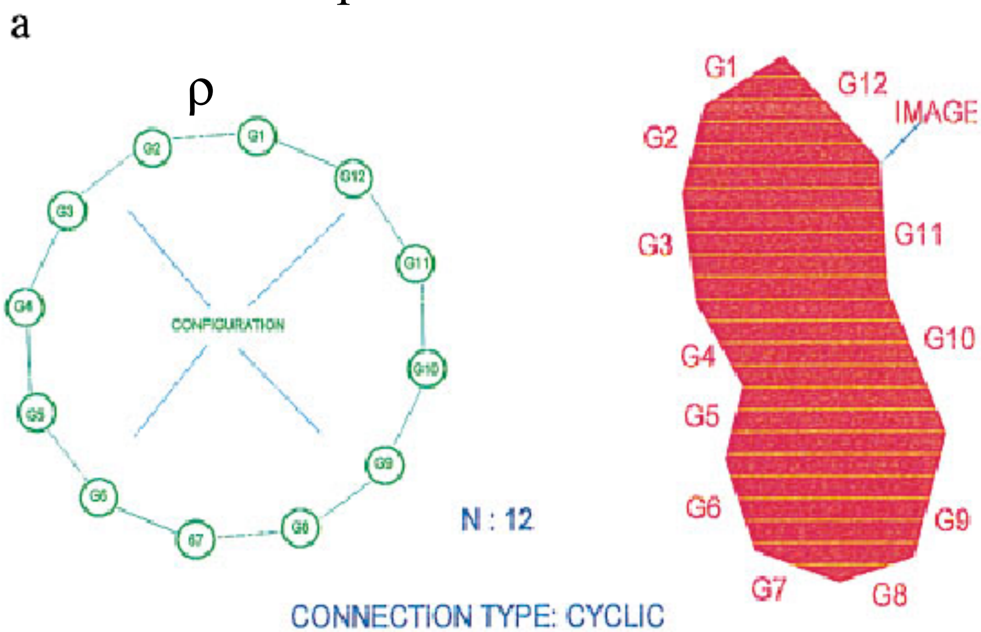
- G --- a set/space of generators (the basic elements in a pattern), each generator has a number of "bonds" that can be connected to neighbors.
- S --- a transformation group (such as similarity transform) for the generators
- ρ--- a set of local regularities (rules for the compatibility of generators and
- Σ--- a set of global configurations (graphs with generators being vertices and connected bonds being edges).

Concept Illustration





Basic description - Points with connections



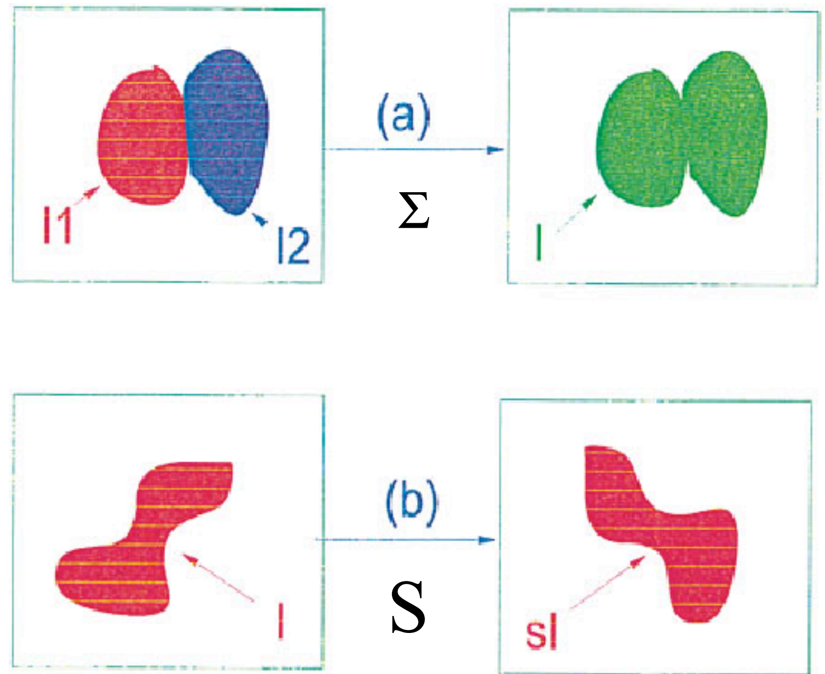


FIG. 3. Algebraic operations. (*a*) Combination. (*b*) Similarity.

Two Schools of Pattern Rec.

- Generative methods:

  Bayesian school, pattern theory. The goal is to learn a complete model for the pattern - one capable of generating new instances.

  1). Specify patterns in terms of measurable signals

  2). Specify likelihood model for how signals are generated from hidden structures

  3). Learning probability models on hidden structures from ensemble of signals

  4). Infer generating causes from particular signals.

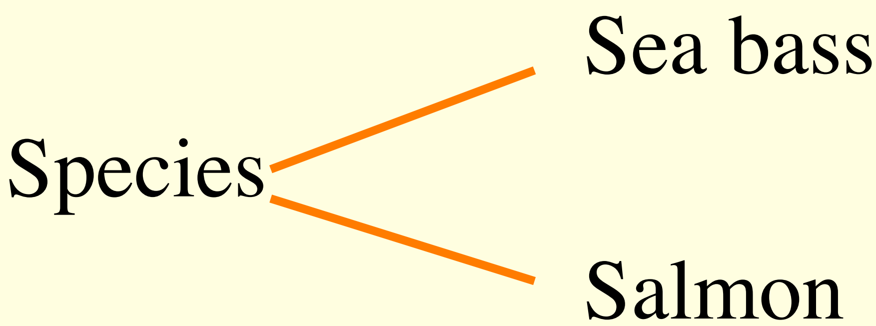
- Discriminative methods:
  - The goal is to tell apart a number of patterns, say 100 people, 10 digits, *directly*, without causes or mathematical description of pattern generating mechanisms.
  - "Don't oversolve your problem".

Pattern Recognition applications

- Build a machine that can recognize patterns:
  - Speech recognition
  - Fingerprint identification
  - OCR (Optical Character Recognition)
  - DNA sequence identification
  - Text Classification

An Example

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

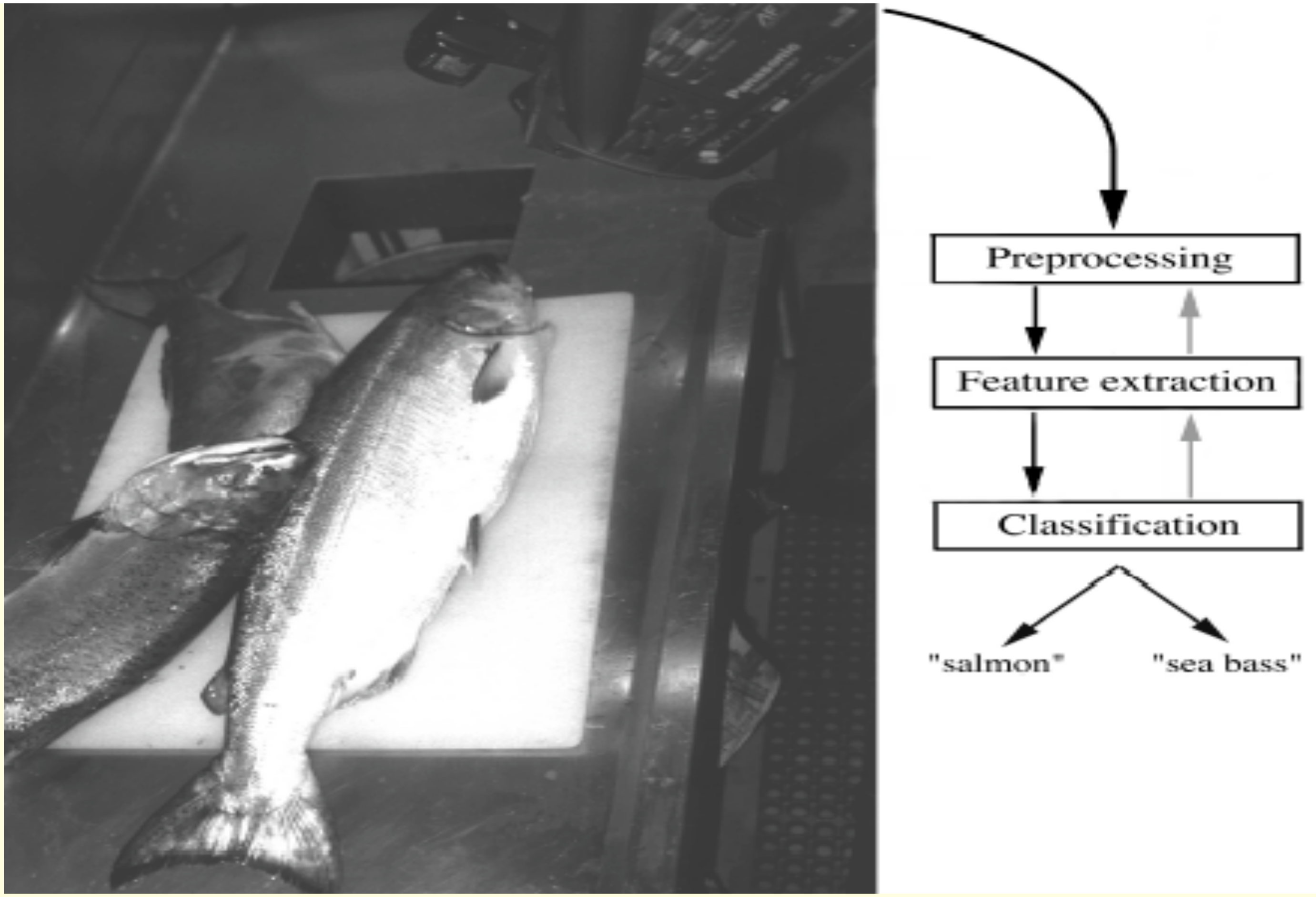


- Problem Analysis

  - 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

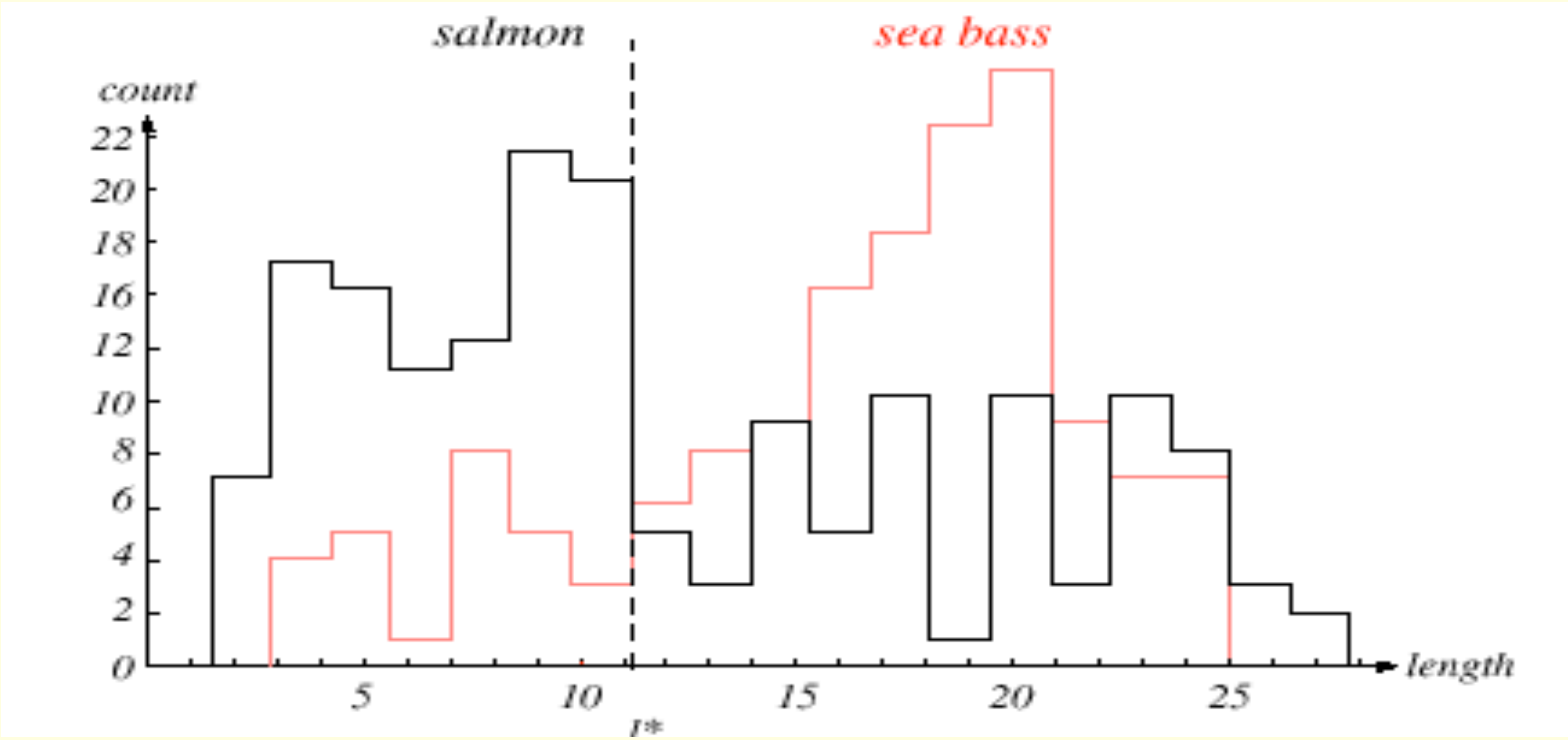
Segment fish from Background

Feature Extraction

Image data from each fish Summarized by feature extractor whose purpose is to reduce the data by measuring certain features

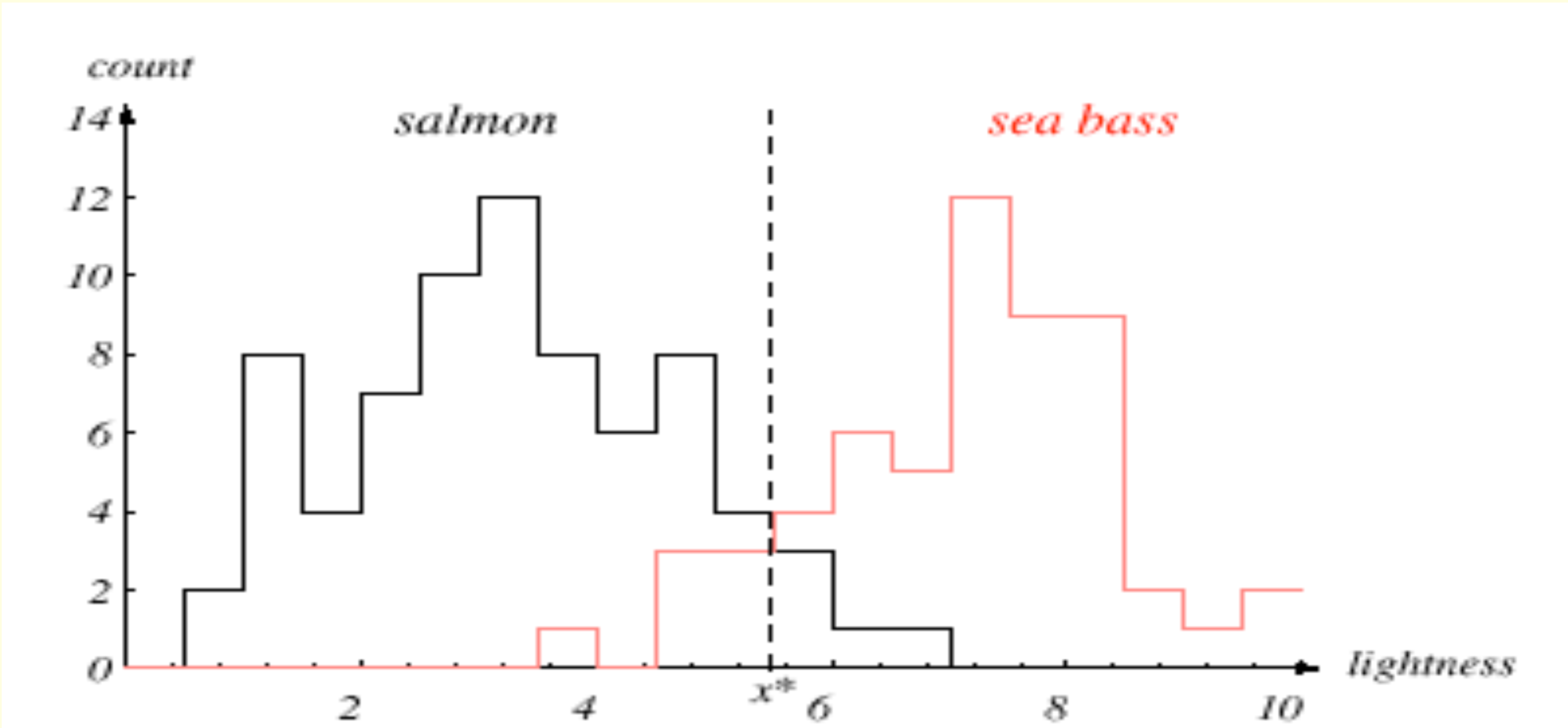
Classification

The features are passed to a classifier, that uses the features to decide which class the instance belongs to.


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

The length is a poor feature alone!

Select the lightness as a possible feature.


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

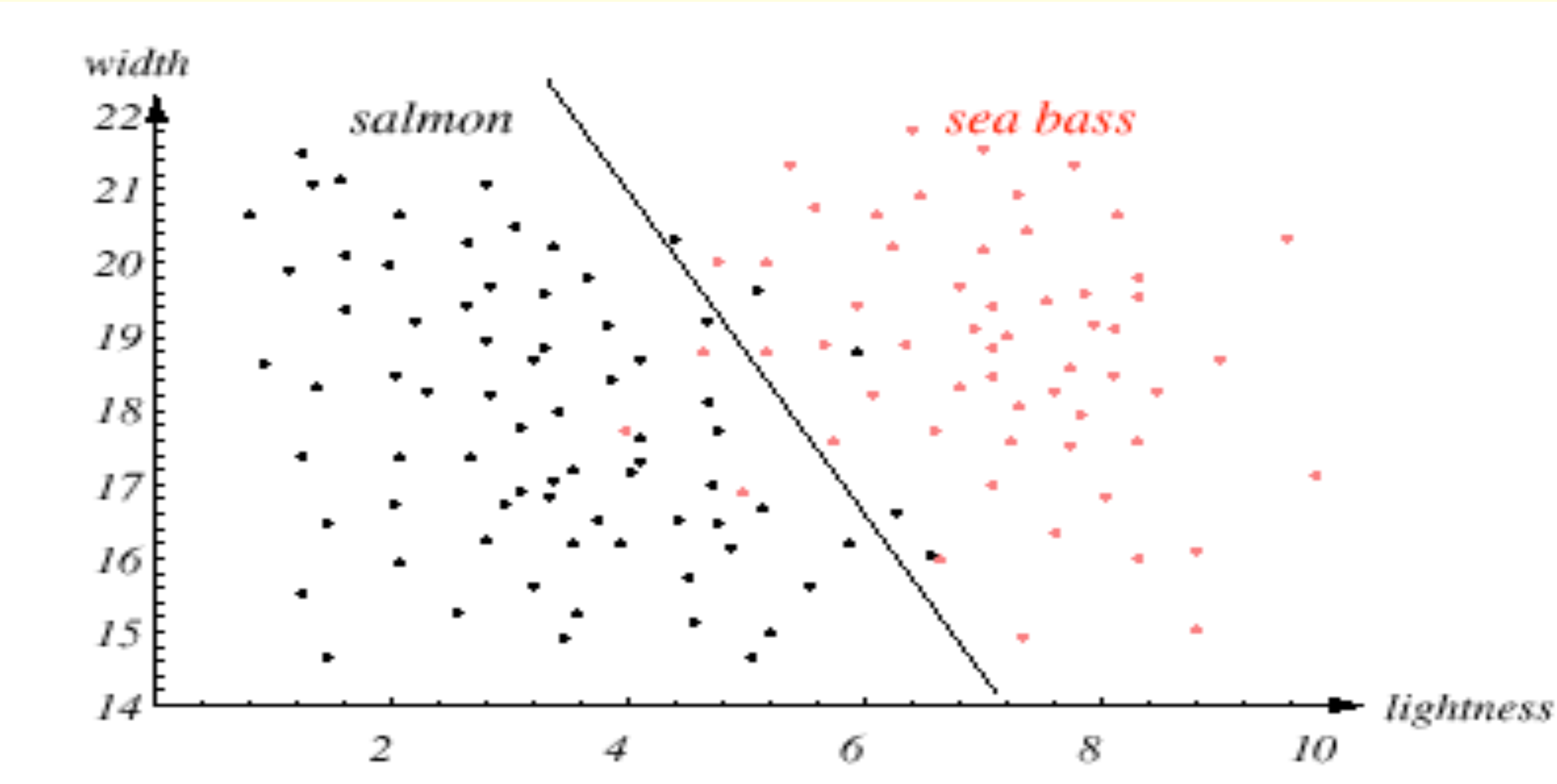
- Threshold decision boundary and cost relationship
  - Move our decision boundary toward smaller values of lightness in order to minimize the cost (reduce the number of sea bass that are classified salmon!)

Task of decision theory

- Adopt the lightness and add the width of the fish

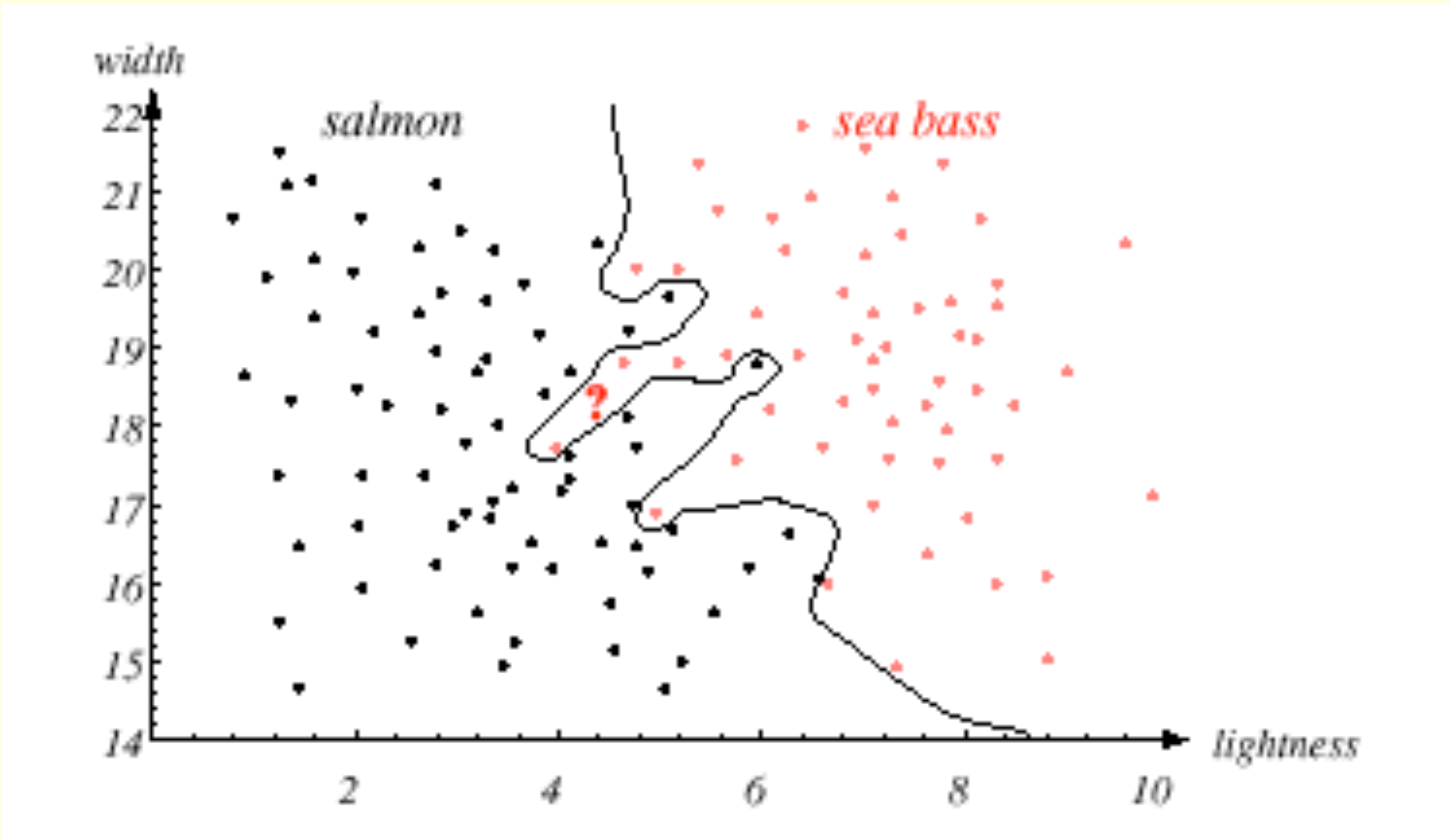
Fish → $x^T = [x_1, x_2]$

x_1 → **Lightness**, x_2 → **Width**


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

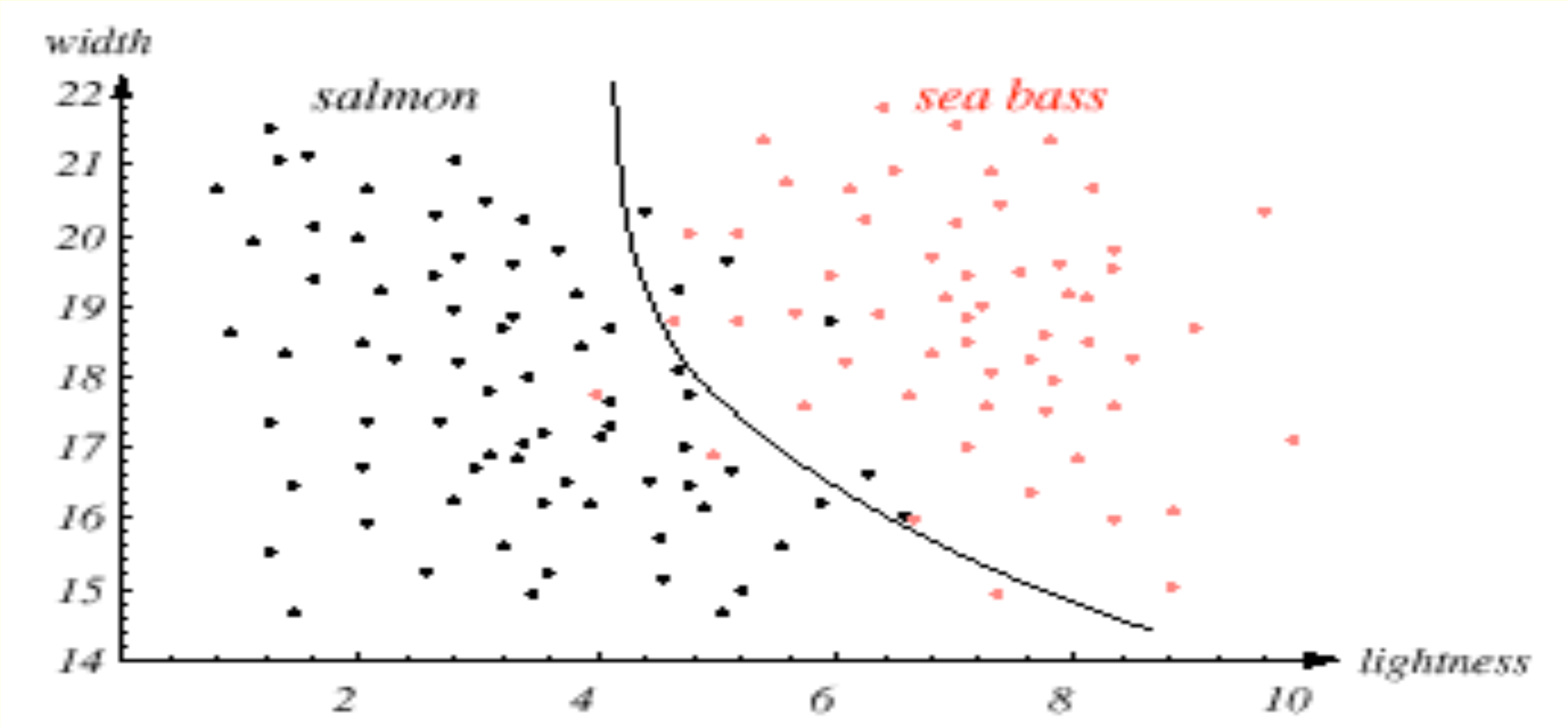
- We might add other features that are not correlated with the ones we already have. A precaution should be taken not to reduce the performance by adding such “noisy features”

- 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

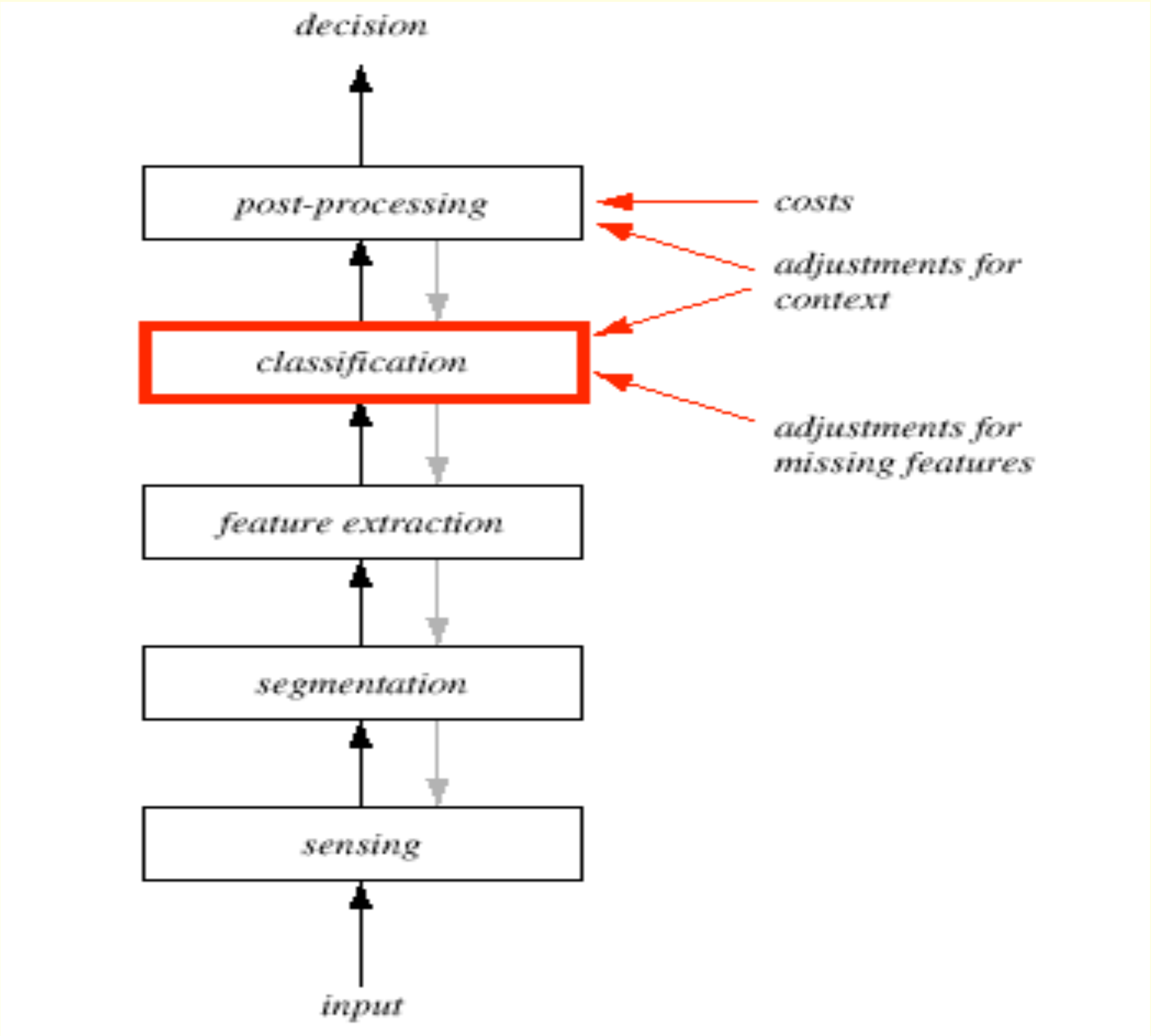
- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel input

Issue of generalization!


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

Pattern Recognition Systems

- Sensing
  - Use of a transducer (camera or microphone)
  - PR system depends of the bandwidth, the resolution sensitivity distortion of the transducer
- Segmentation and grouping
  - Patterns should be well separated and should not overlap


decision
post-processing
costs
adjustments for context
classification
adjustments for missing features
feature extraction
segmentation
sensing
input

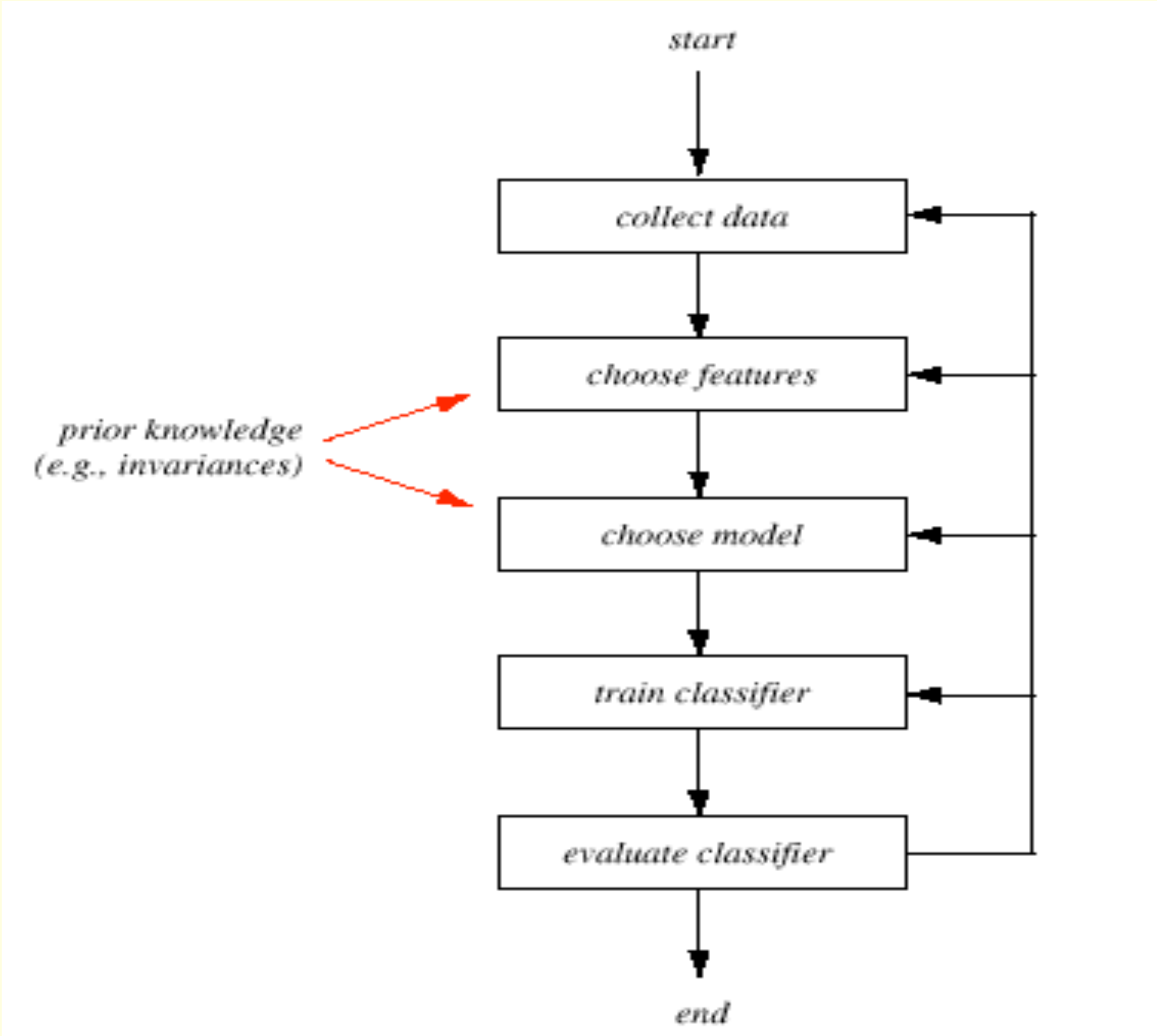
- Feature extraction
  - Discriminative features
  - Invariant features with respect to translation, rotation and scale.

- Classification
  - Use a feature vector provided by a feature extractor to assign the object to a category

- Post Processing
  - Exploit context: input dependent information other than from the target pattern itself to improve performance

The Design Cycle

- Data collection
- Feature Choice
- Model Choice
- Training
- Evaluation
- Computational Complexity


start
collect data
choose features
prior knowledge
(e.g., invariances)
choose model
train classifier
evaluate classifier
end

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

- Feature Choice
  - Depends on the characteristics of the problem domain.
  - Feature desirata:
    - Simple to extract,
    - invariant to irrelevant transformations
    - insensitive to noise.

- Model Choice

    - 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

- 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 and Adaptation

- Supervised learning
  - A teacher provides a category label or cost for each pattern in the training set
- Unsupervised learning
  - The system forms clusters or “natural groupings” of the input patterns
- Reinforcement Learning/Semi-supervised
  - Environment provides feedback indicating the quality of decisions, but no labeled examples.

Conclusion

- The number, complexity and magnitude of the sub-problems of Pattern Recognition are formidable.

- Many of these sub-problems can indeed be solved

- Many fascinating unsolved problems still remain