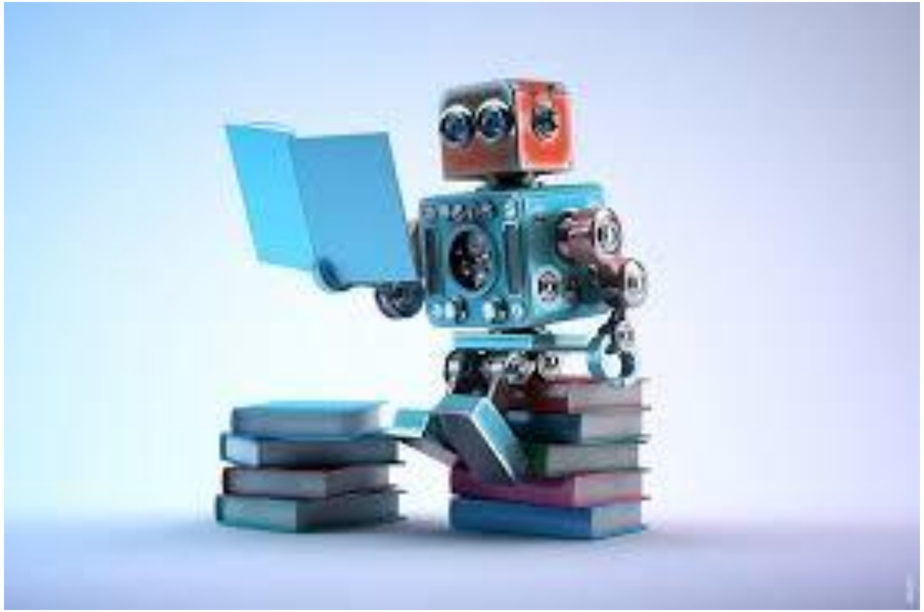
Machine Learning – CS643

Dr. Sheikh Faisal Rashid

- Assistant Professor: Computer Science, UET Lahore
- Director: Artificial Intelligence Research Lab (AIRL), KICS
- Vice President: Pakistan Pattern Recognition Society (PPRS)
- Visiting Researcher DFKI, Germany
- shfaisal@gmail.com



Department of Computer Science and Engineering UET, Lahore

Overview

- Pattern Classification
  - Introduction
  - Concepts
  - Design Cycle

Pattern Classification

- More than a program
  - Usually, we think a program is something written by an experienced person.

Pattern Classification

- More than a program
  - Usually, we think a program is something written by an experienced person.
  - Often, the program isn't complete without "experience" of its own.

Pattern Classification

- More than a program
  - Usually, we think a program is something written by an experienced person.
  - Often, the program isn't complete without "experience" of its own.
  - The idea of writing programs that use data (experience) to create better programs than people can write directly.

Pattern Classification

- Pattern classification systems make decisions
- Decisions are usually made autonomously
- Decisions are not pre-programmed
- Decision “rules” are derived from data

SE, AI, PR

● Software Engineering
○ Manual creation of specifications, manual implementation, full control over details of execution.
● Artificial Intelligence (Rule-Based Systems)
○ Manual creation of specifications, specifications are directly executable (rule interpreters, etc.). Details of execution are automated.
● Pattern Recognition, Machine Learning
○ Programmer chooses category of application, but detailed specifications are automatically derived from data. Execution is automated.

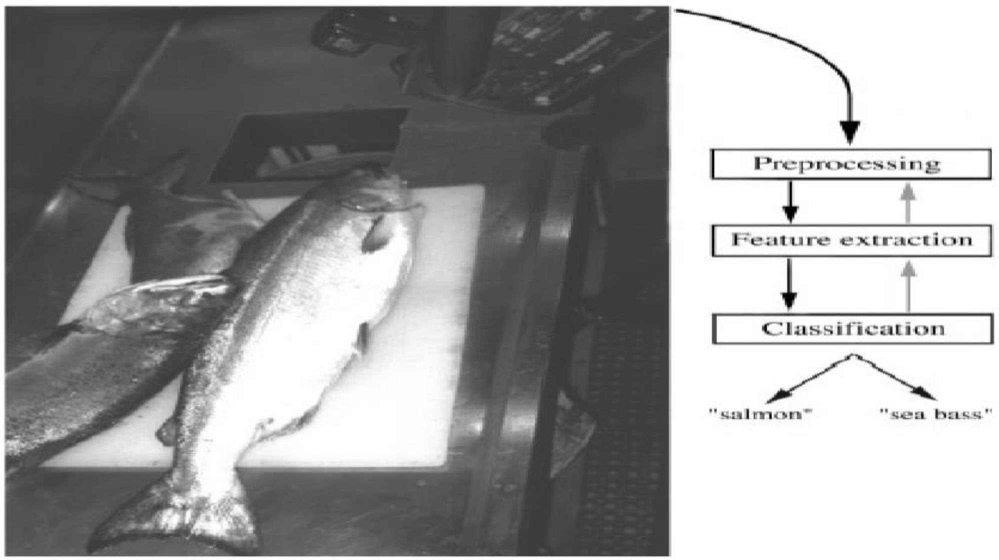
Fish Classification: An example

- "Sorting incoming Fish on a conveyor according to species using optical sensing"
- Species
  - Sea bass
  - Salmon

Problem Analysis

- set up a camera and take some sample images to extract features
- feature types
  - length (positive real number)
  - lightness (positive real number)
  - width (positive real number)
  - number of fins (non-negative integer)
  - shape of fins (one-of a set of possible categories)
  - position of the mouth (one of a set of possible categories)
  - ...
- base the decision of which kind of fish it is on these measurements

Overall classifier



Preprocessing

- raw camera image may be 1024 x 1024 pixels
  - >1 million numbers
- contains lots of irrelevant data
  - background
  - dirt
  - …
- feature extraction
  - data reduction—computational efficiency
  - remove irrelevant variation
- feature measurements are passed to the classifier

Feature Vector

- collection of measurements like an “object” or “structure” or “database record”
- example

{ length = 21cm,

lightness = 0.73,

width = 8.3cm,

number_of_fins = 3,

shape_of_fins = {square, triangular, square},

position_of_the_mouth = {front} }

Dataset

like a spreadsheet table (with millions of rows...)

- training data: class + features
- test data: features only

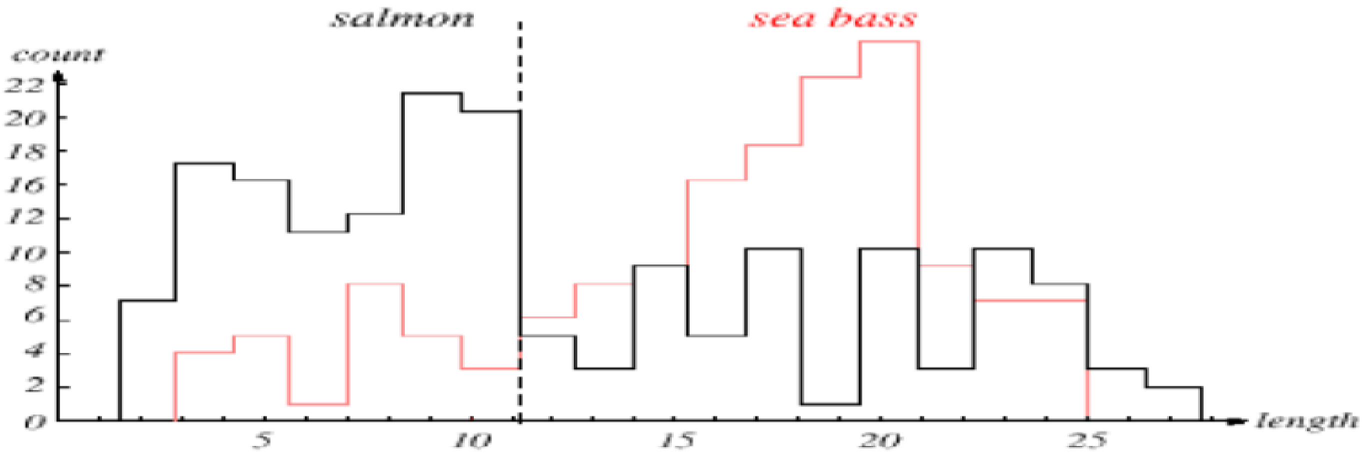
	A	B	C	D	E	F	G
1	class	length	lightness	width	#fins	shapes	mouth
2	2	28.59	0.35	15.99	2	s,s	f
3	2	22.18	0.37	13.31	1	t	b
4	1	16.29	0.28	16.89	2	t,t	f
5	2	29.87	0.46	14.49	1	t	b
6	2	27.65	0.43	13.11	2	s,s	f
7	2	29.75	0.57	13.70	2	s,s	f
8	2	26.43	0.30	13.54	1	s	f
9	1	24.13	0.25	12.50	1	t	f
10	2	27.25	0.47	14.05	1	t	f
11	2	24.13	0.32	12.23	1	t	f
12	2	22.48	0.58	14.14	1	t	f
13	1	21.14	0.48	16.16	1	t	f
14	1	16.11	0.44	15.75	2	t,t	f
15	1	21.26	0.34	12.71	2	t,t	f
16	2	28.54	0.51	17.00	1	s	f
17	1	18.99	0.25	12.31	1	s	f
18	2	26.98	0.39	16.50	2	t,t	f
19	1	20.50	0.29	16.56	2	t,s	f
20	1	22.67	0.45	15.61	2	t,t	f
21	1	17.20	0.37	16.03	2	t,s	b
22	1	21.38	0.26	12.08	2	t,t	f
23	1	20.73	0.41	14.93	2	t,s	f

Histogram

frequency of each feature

- "binned" for real-valued features

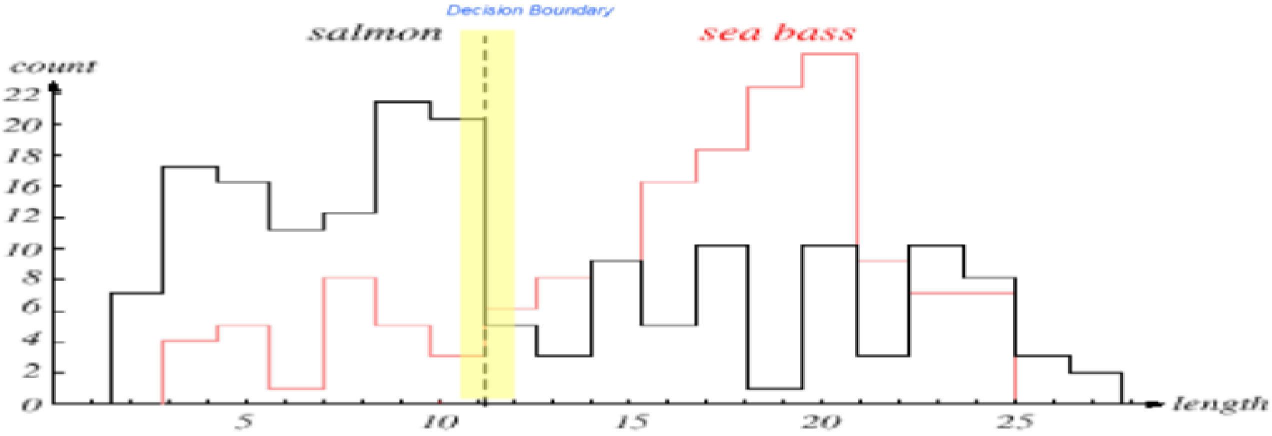
let us guess a classifier by eye



Decision Rule

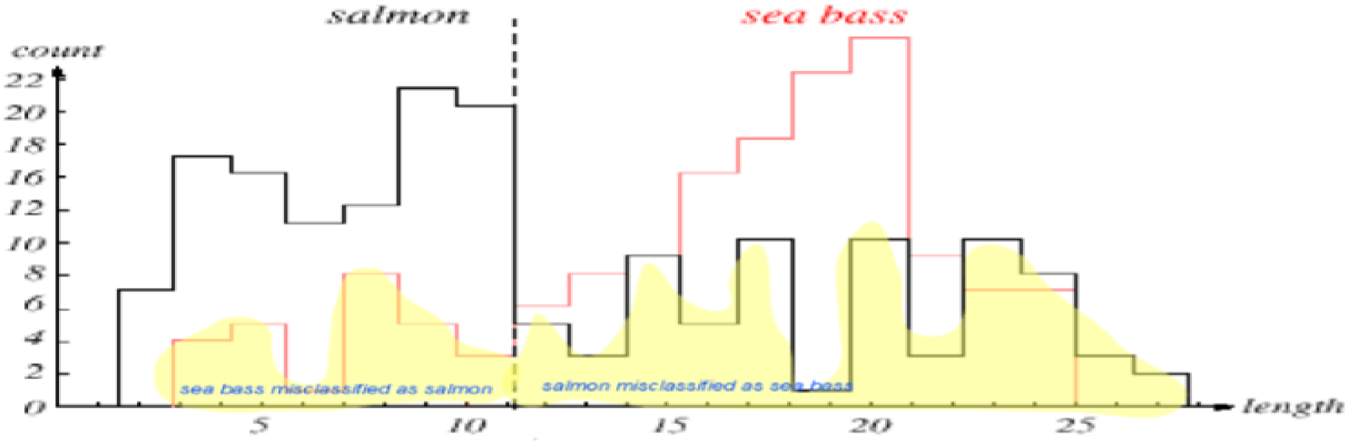
A Decision Rule is a function mapping feature vectors into classes

Decision(features) =
IF features.length<11 THEN return "salmon"
ELSE return "sea bass"



Empirical Error Rate

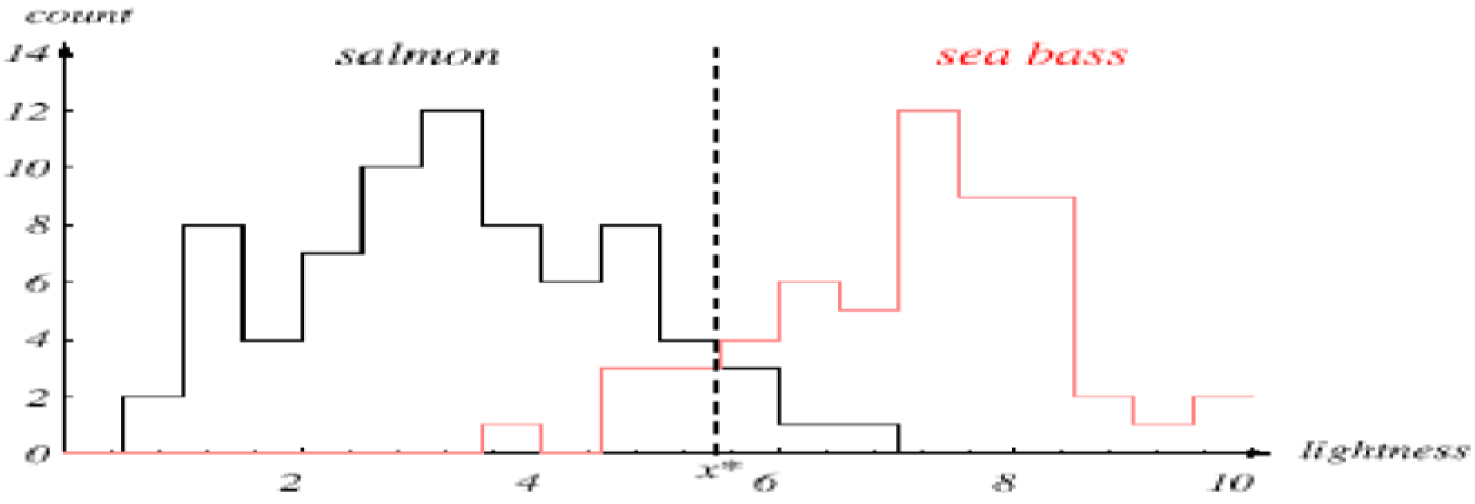
fraction of misclassified samples among total samples
"empirical" because it is estimated from a data sample



Finding Good Features / Boundaries

using lightness, we can separate the two classes better than using length

moving the decision boundary changes the error rate



Decision Theory

- 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

Multiple Features

- Adopt the lightness and add the width of the fish

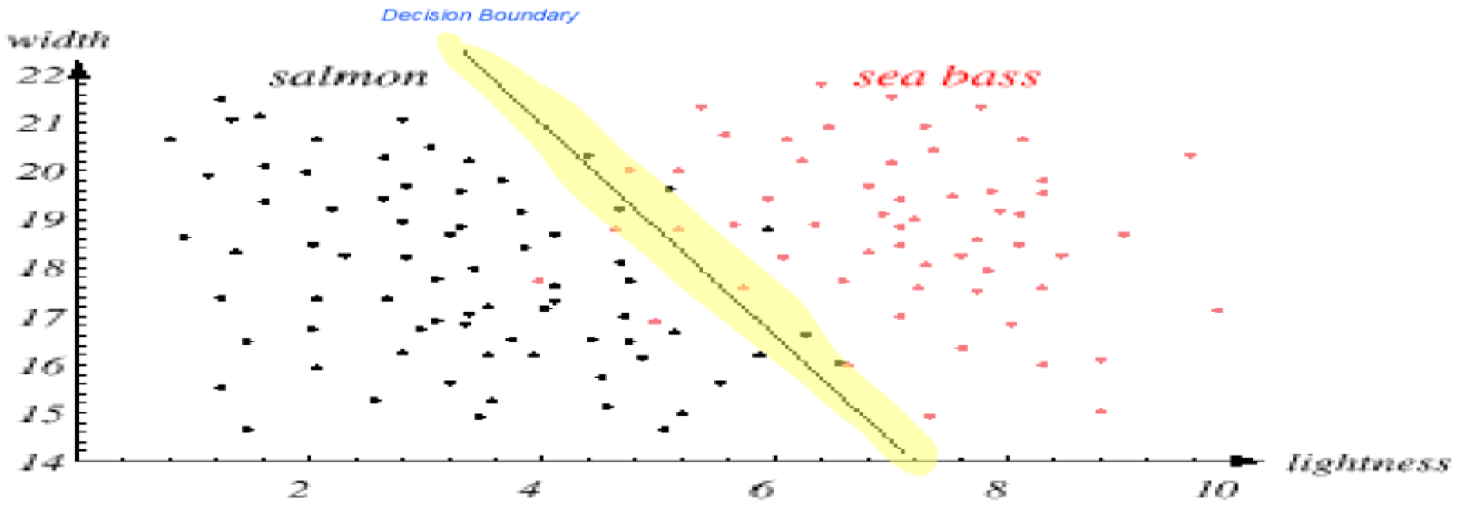
Fish $\Longrightarrow$ $x^T = [x_1, x_2]$

Lightness

Width

Multiple Features

Scatterplot



Linear Decision Function

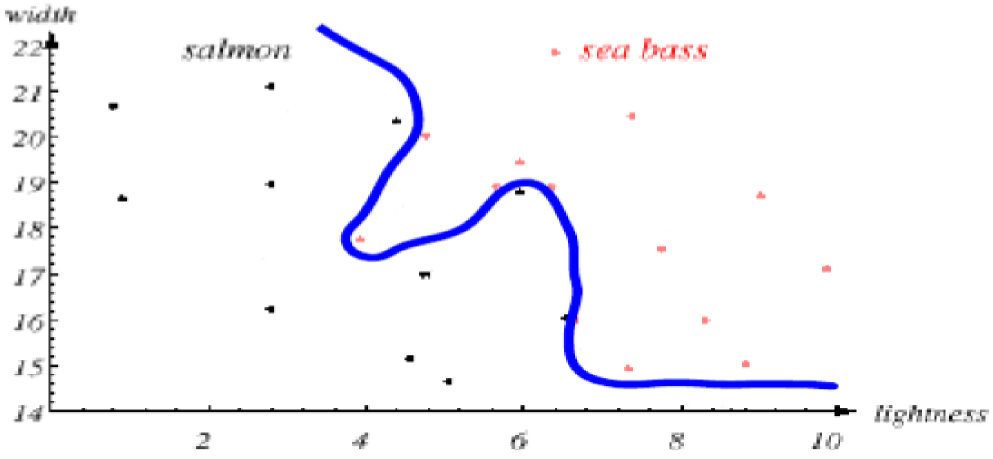
```
Decision(features) =
  IF features.lightness * 0.6 + features.width * 0.11 – 21.3 < 0
    THEN return "salmon"
    ELSE return "sea bass"
```

decision function depends on two numerical features

feature values enter linearly into the decision function

Generalization

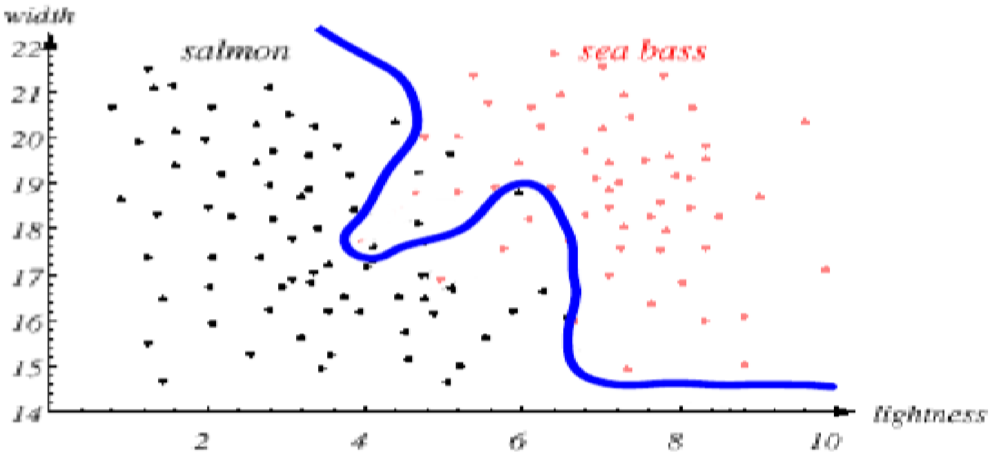
decision boundaries that work perfectly on a few samples...



Overfitting

… may fail to generalize well to new data (overtraining)

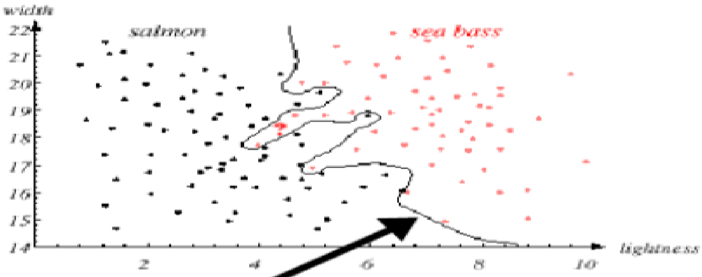
ensuring good generalization is a key problem in pattern recognition



Avoid overtraining

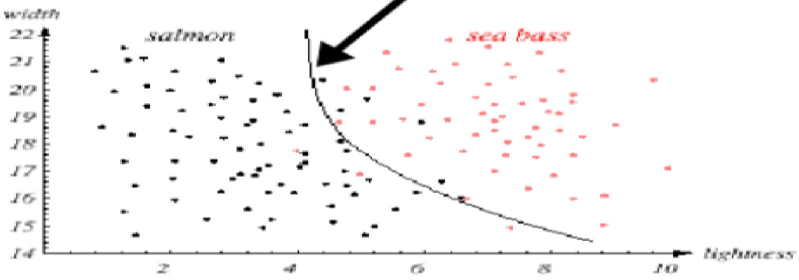
choose a tradeoff between...

- performance on the training data
- "complexity" of the classifier / decision boundary

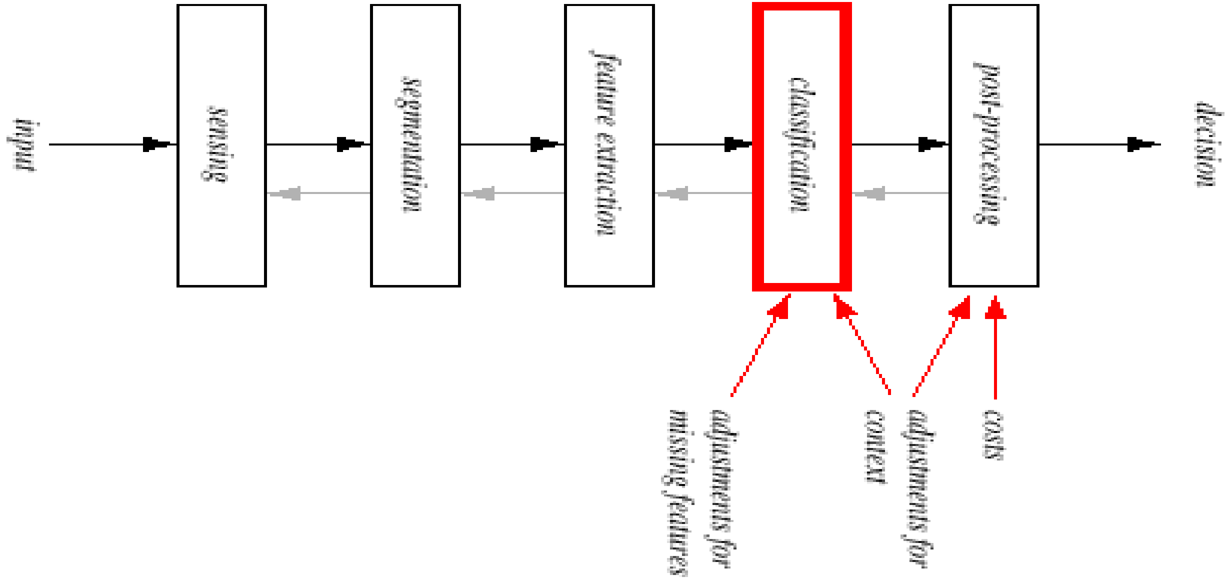


decision boundary is too complex

probably a good tradeoff



Pattern Recognition Systems



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

Pattern Recognition Systems

- 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
- Model Selection
- Computational Complexity

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?

The Design Cycle

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

The Design Cycle

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

The Design Cycle

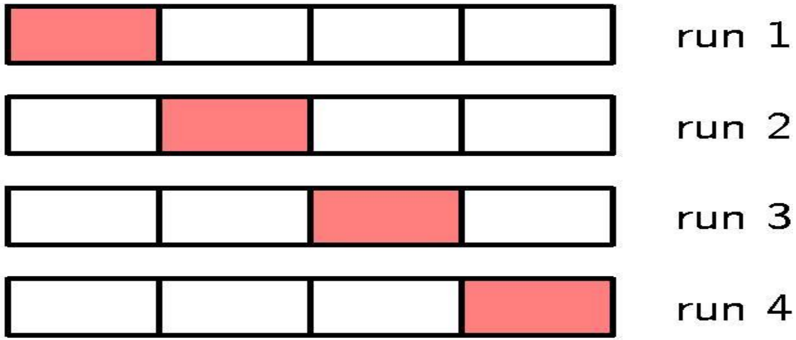
- Training
  - Use data to determine the classifier.
  - Many different procedures for training classifiers and choosing models

The Design Cycle

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

The Design Cycle

- Cross-Validation



The Design Cycle

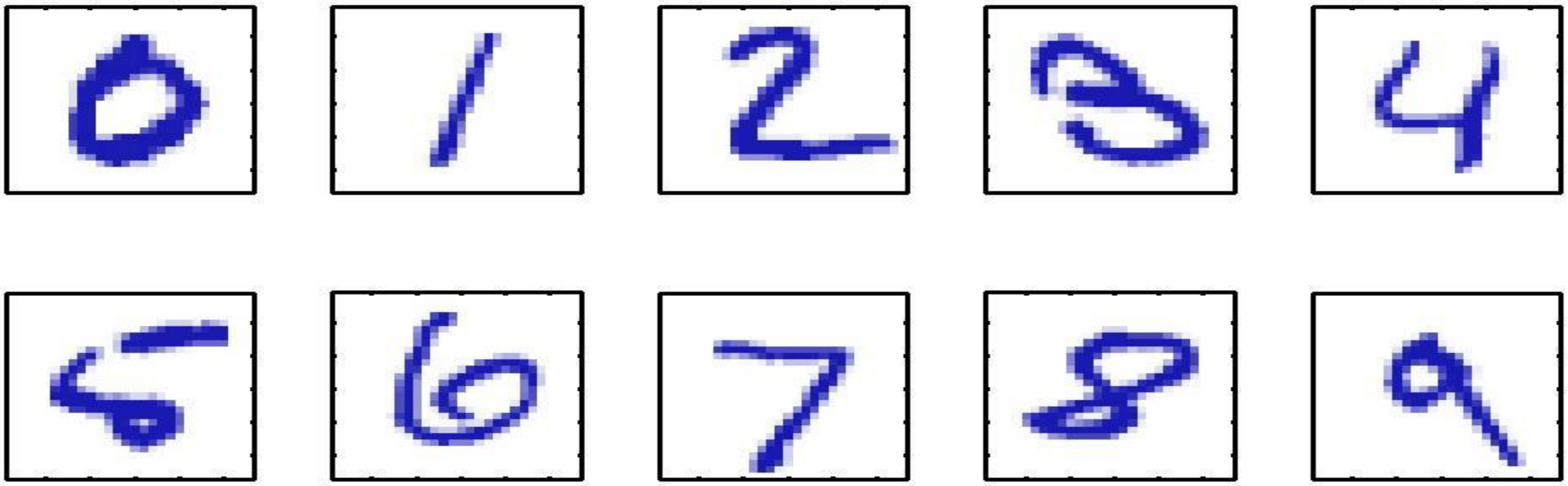
- Computational Complexity
  - What is the trade-off between computational ease and performance?
  - (How an algorithm scales as a function of the number of features, patterns or categories?)

Important terminology

- Training data
  - Data samples
  - Target vectors
- Learning / Training
  - Machine takes training data and automatically learns mapping from data samples to target vectors
- Test data
  - Target vectors are concealed from the machine
  - Machine predicts the target vectors based on previously learned model
  - Accuracy can be evaluated by comparing the predicted vectors to the actual vectors

Example

Handwritten Digit Recognition



Problem Statement

- Consider a 28 x 28 pixel image
- Represented by a 784 dimensional vector **x**
- Goal: build a machine that takes the vector **x** as input and produces the identity of digit 0,…,9 as the output

Machine Learning

- Tom Mitchel's definition of Machine Learning:
  A computer program is said to learn from experience ***E*** with respect to some class of tasks ***T*** and performance measure ***P***, if its performance at tasks in ***T***, as measured by ***P***, improves with experience *E*.

Examples

Example 1: A Chess learning problem

- Task T: playing chess
- Performance measure P: percent of games won against opponents
- Training Experience E: playing practice games against itself

Examples

Example 2: Autonomous Vehicle Problem

- Task T: driving on a public highway/roads using vision sensors
- Performance Measure P: percentage of time the vehicle is involved in an accident
- Training Experience E: a sequence of images and steering commands recorded while observing a human driver