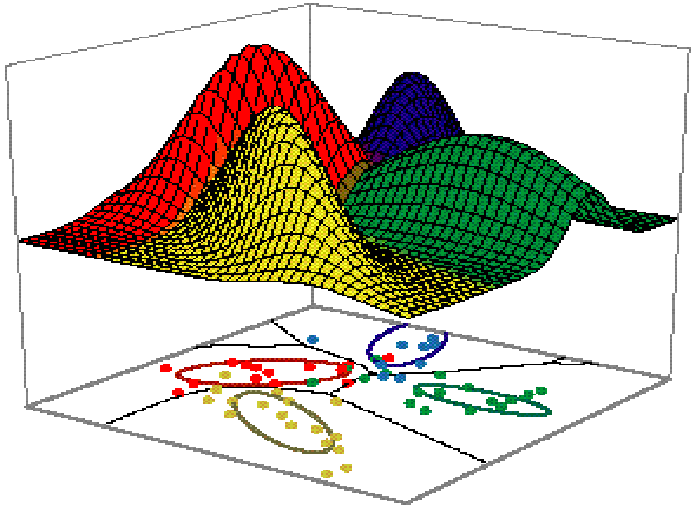
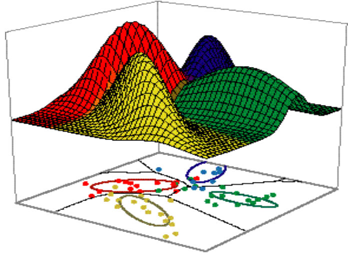
Part 1: Introduction to Pattern Recognition



Machine Perception
The Big Picture
An Example
Pattern Recognition Systems
The Design Cycle
Learning and Adaptation
Course Objectives

Some materials in these slides were taken from Pattern Classification (2nd ed) **by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000; Chapter 1**

Machine Perception

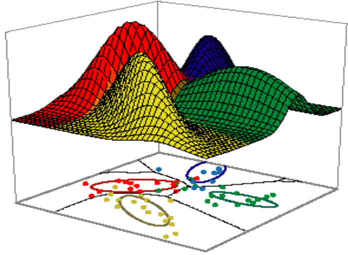


- Humans naturally recognize patterns
  - Feel objects in your pocket
  - Judge ripeness of an apple by smell
  - Guess at emotion from facial expression
  - Reading handwriting

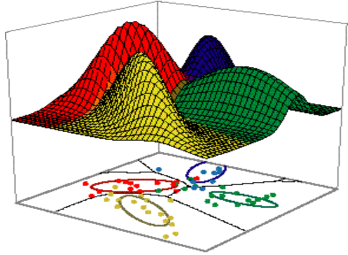
- These are all extremely difficult for a machine!

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

The big picture (supervised learning)

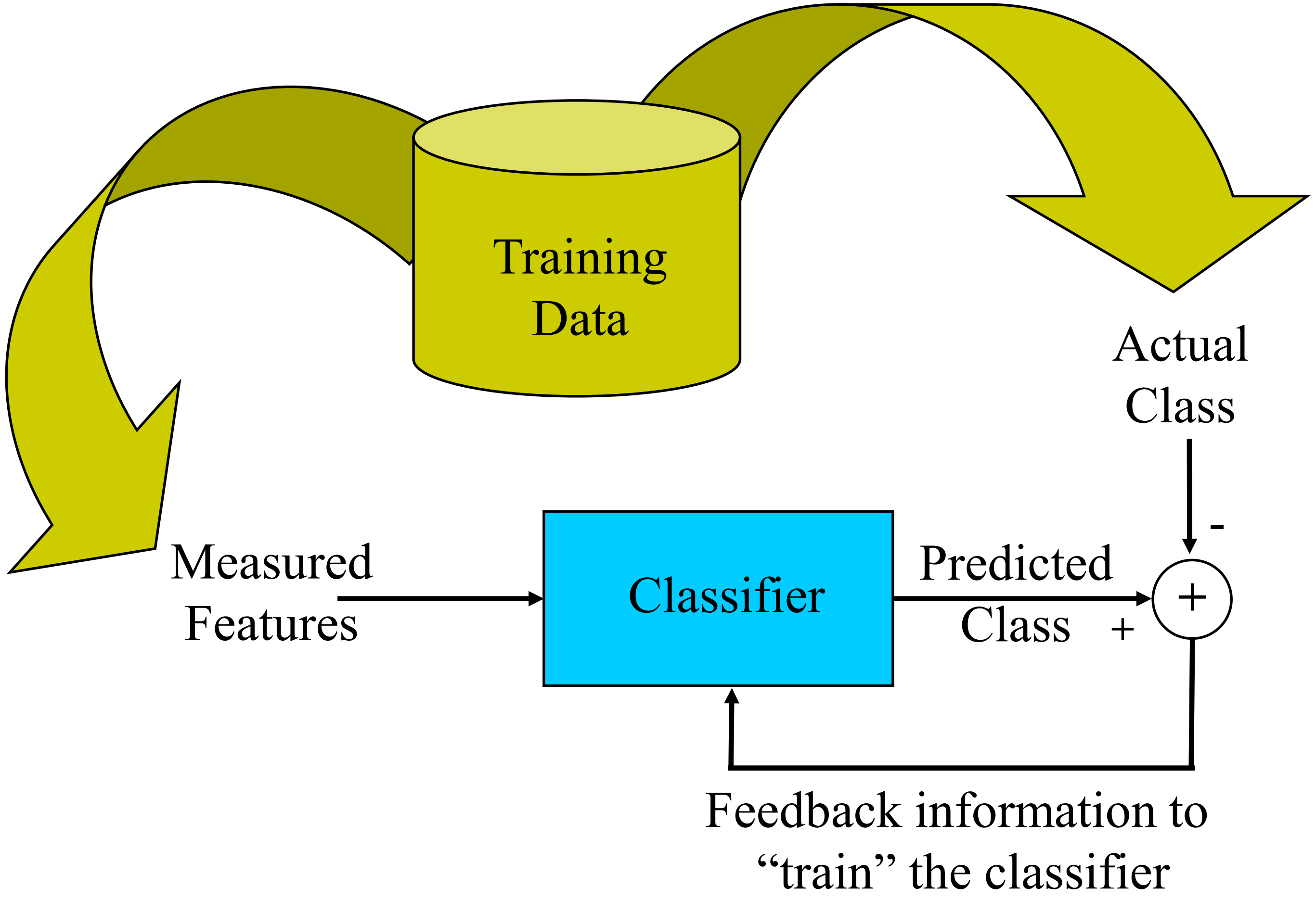


- Training
  - Collect some training samples where the class is known
  - Make some measurements to extract features
  - Train a classifier using measured features and known class
- Testing
  - Evaluate the accuracy of the classifier on test data that was not used to train the classifier.
- Operation
  - Ultimately, system will work for NEW data
  - i.e. examine features for a new sample, guess at class

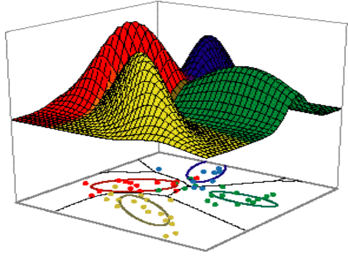


The big picture (supervised learning)

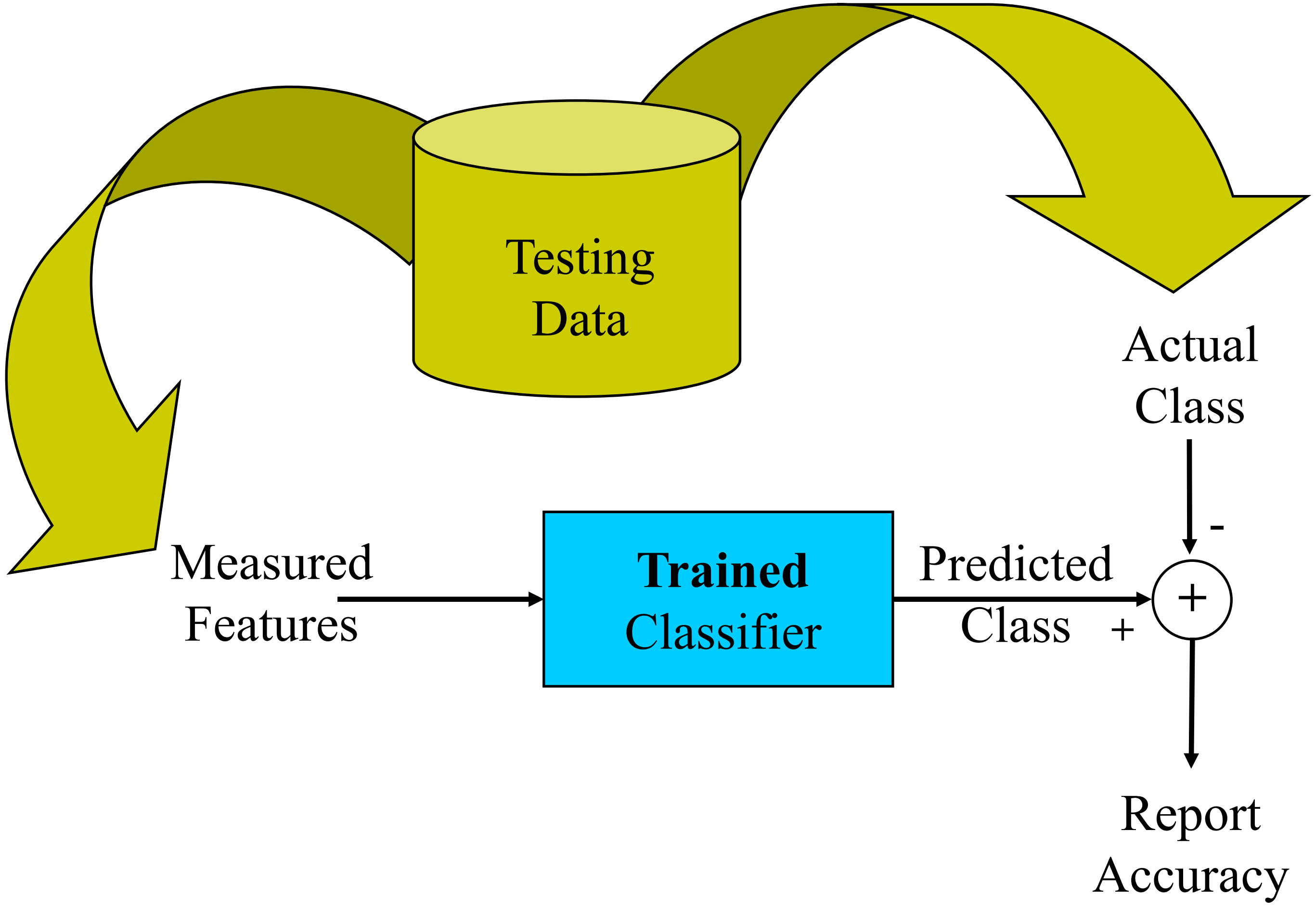
- Training:

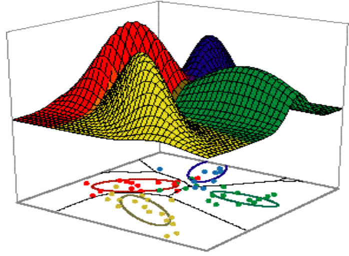


The big picture (supervised learning)



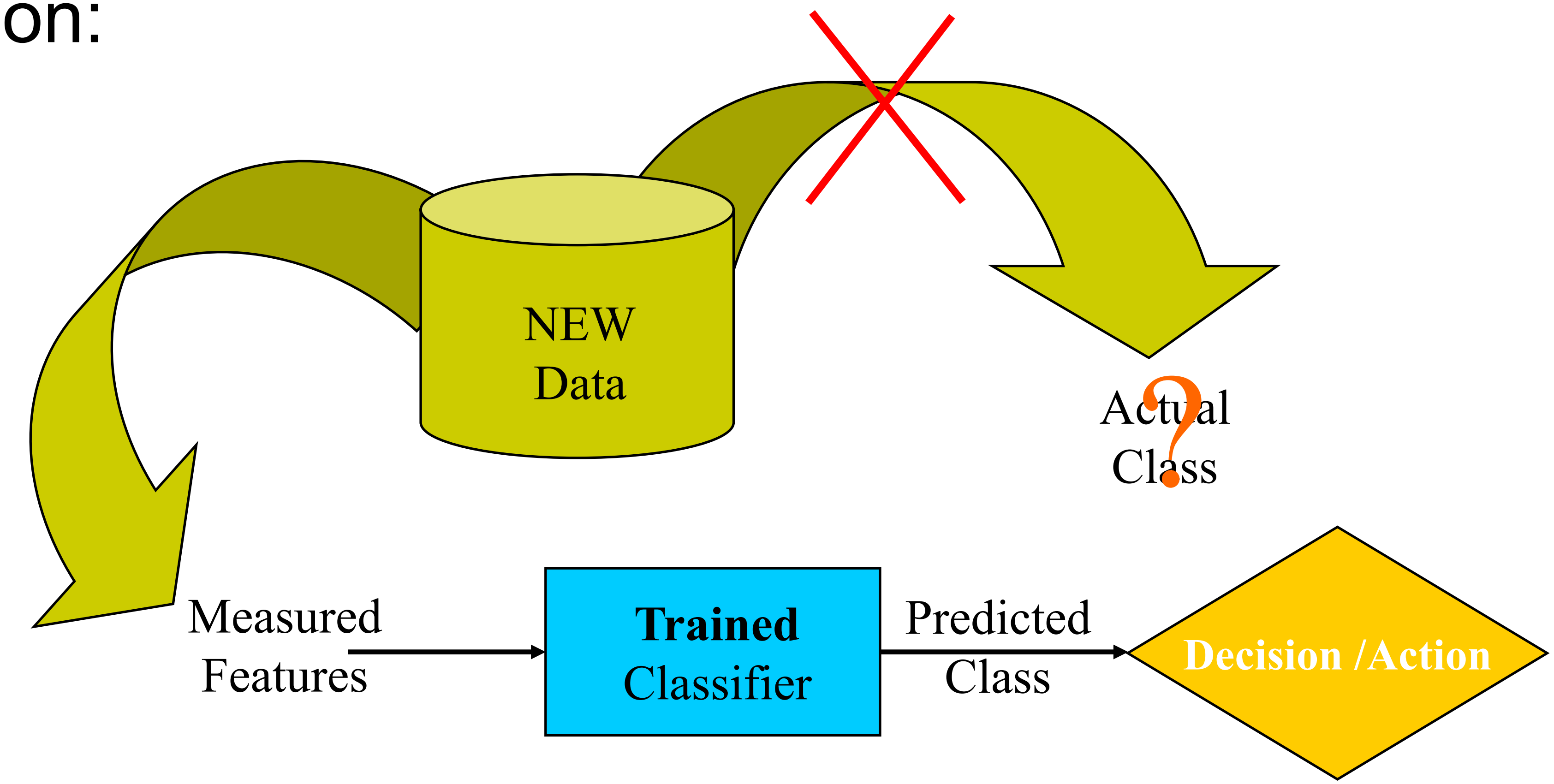
- Testing:

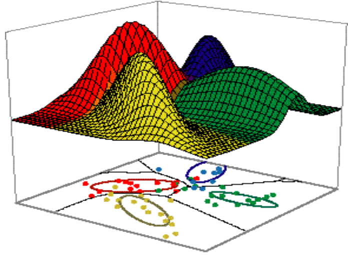




The big picture (supervised learning)

- Operation:





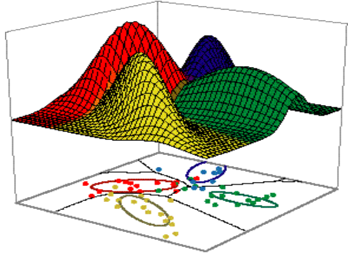
An Example – fish sorter

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

Species → Sea bass

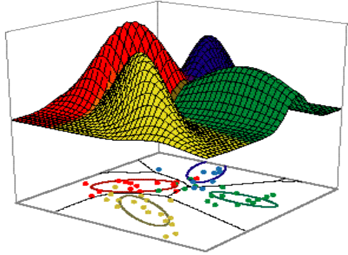
Species → Salmon

An Example – fish sorter



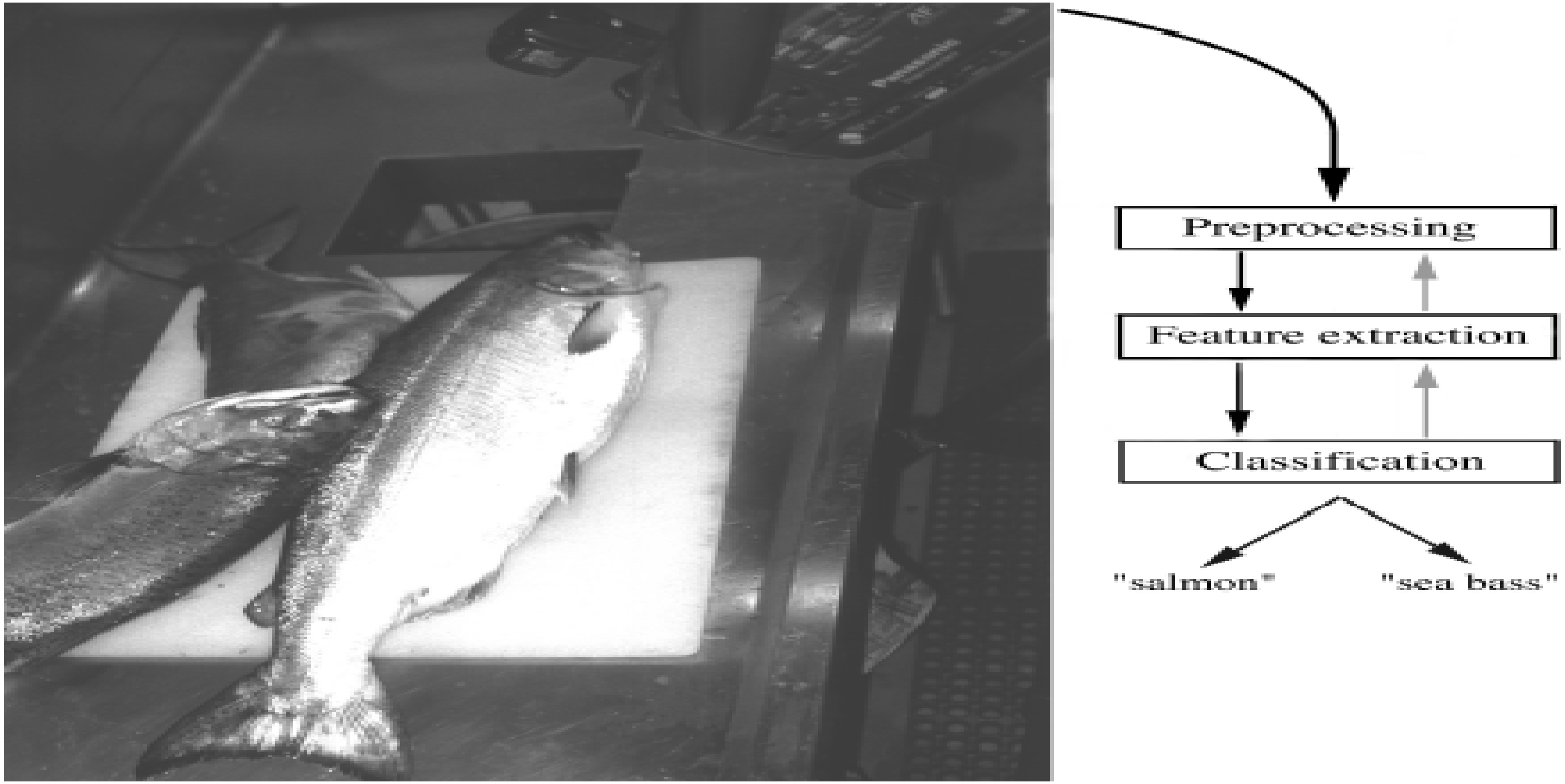
- Problem Analysis
  - Set up a camera and take some sample images to extract features
    - Length
    - Lightness
    - Width
    - Number and shape of fins
    - Position of the mouth, etc…
  - May be continuous, nominal/categorical, ordinal
  - We may use only a subset of these features in our classifier!

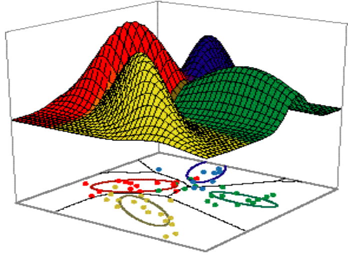
An Example – fish sorter



- 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
  - ‘Features’ vs. ‘measurements’
- The features are passed to a classifier

An Example – fish sorter

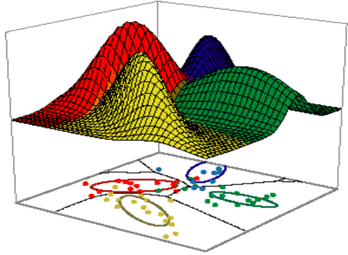


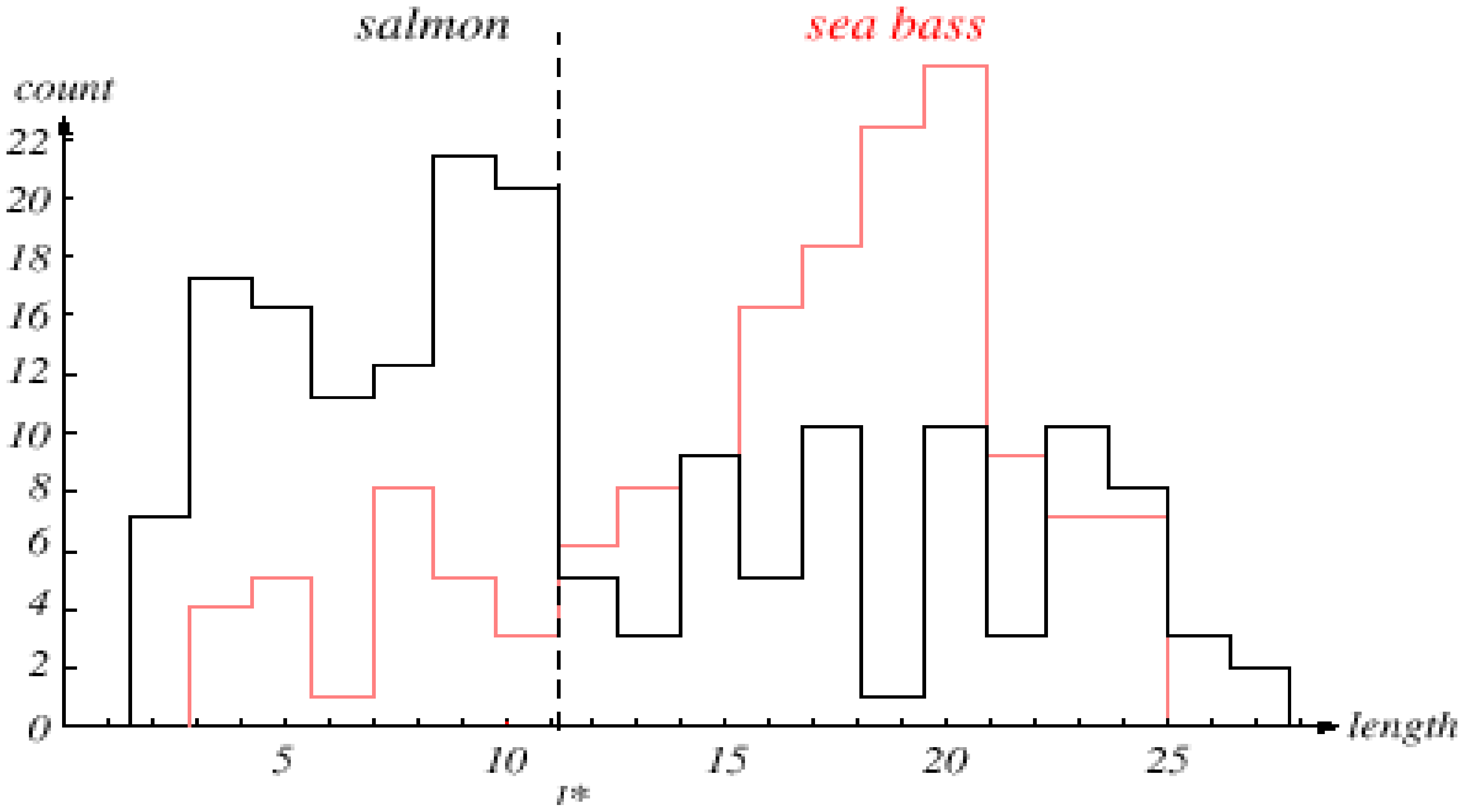


An Example – fish sorter

- Classification
  - Get some prior information:
    - Told that salmon are generally shorter than sea bass
  - Select the length of the fish as a possible feature for discrimination

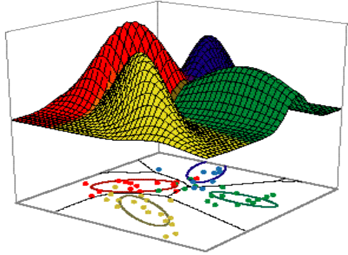
Histogram of fish length





L*, Optimal decision boundary placement

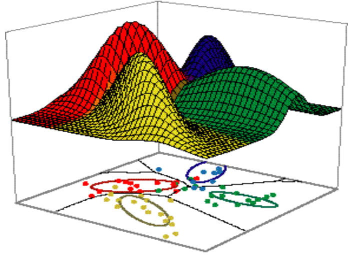
An Example – fish sorter

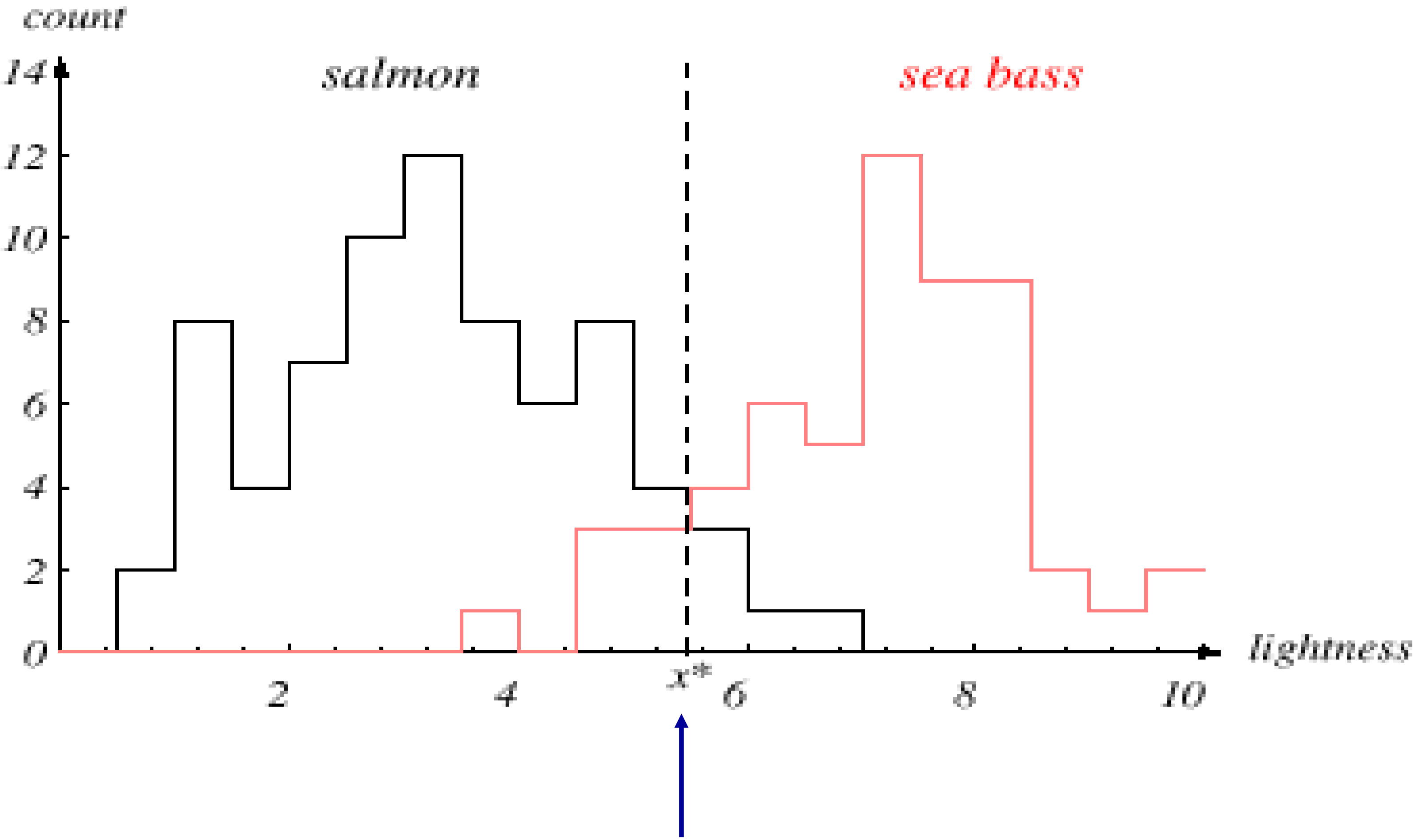


- Although, on average, salmons are shorter than sea bass, length is a poor feature alone!

- Try selecting lightness as a possible feature.

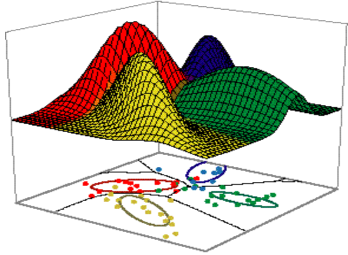
Histogram of fish lightness





x^*, Optimal decision boundary placement

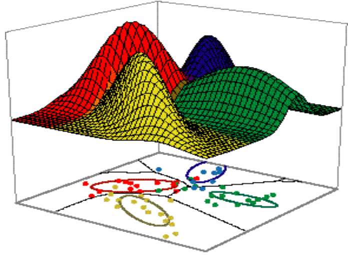
Threshold decision boundary & cost relationship



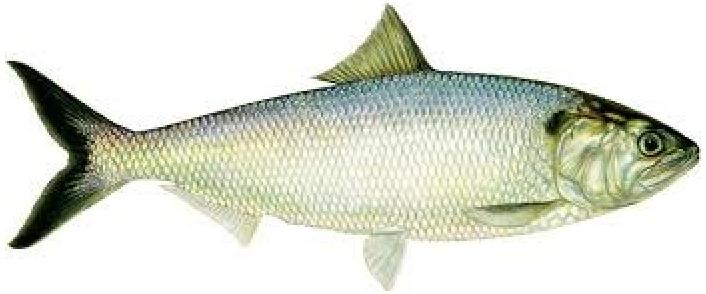
- Not all errors are equal
  - Some misclassifications can have higher costs
  - E.g. missed cancer diagnosis (false negative/missed positve) vs. false positive diagnosis (false positive)
- In this case, selling salmon as sea bass is not a problem, but selling sea bass as salmon will aggravate customers!
  - 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

A new feature vector



- No single feature provides a good separation of the two fish types (classes)
- Try combining multiple features:
  - Adopt the lightness and add the width of the fish

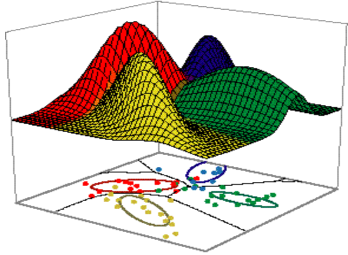


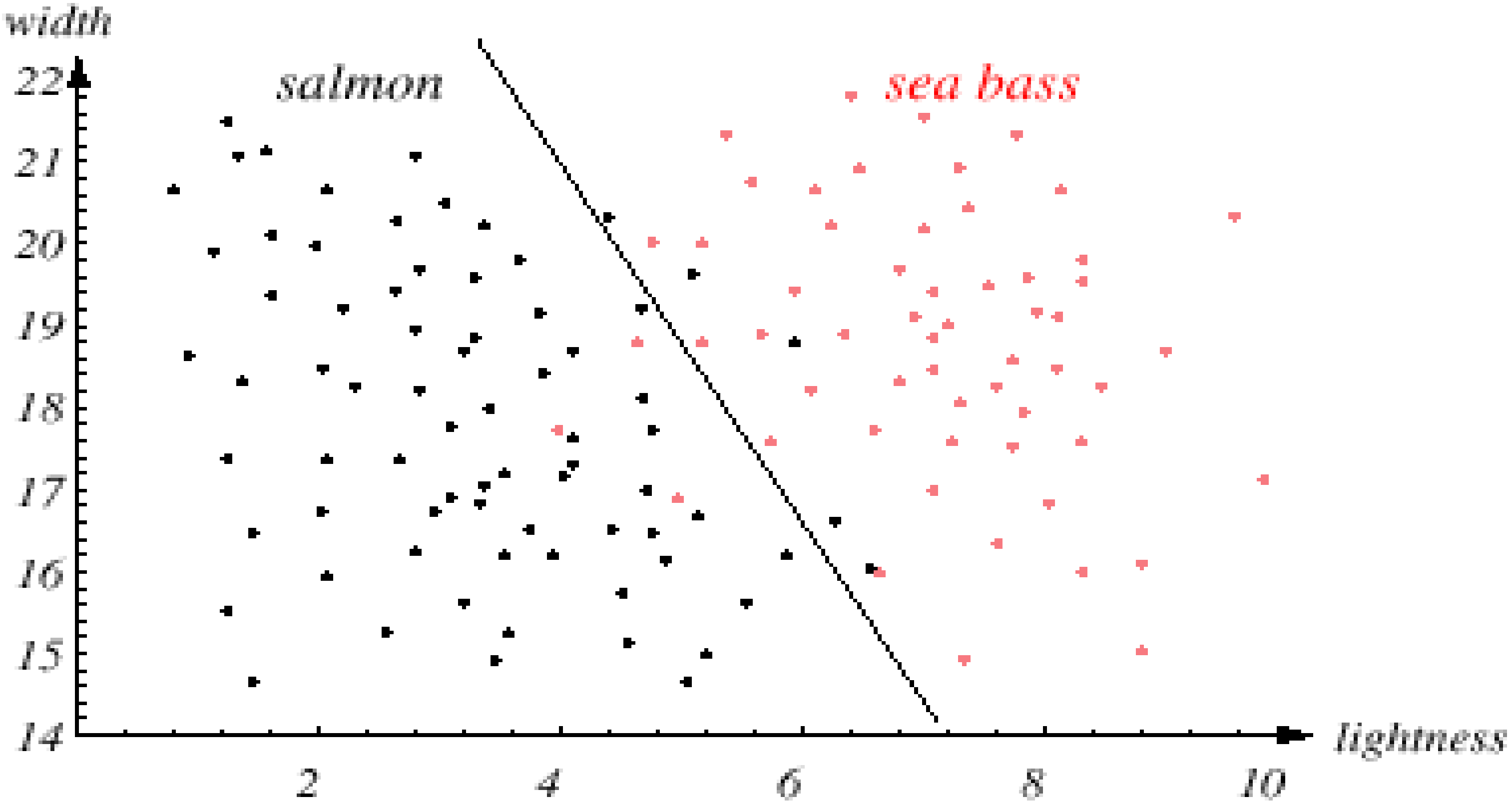
$$\boldsymbol{x}^T = [x_1, x_2]$$

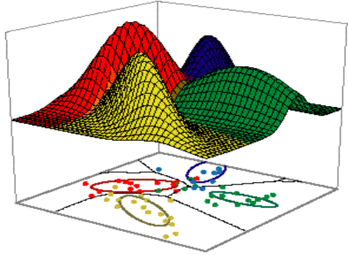
Lightness **Width**

The real object 'fish' is now represented by a <u>feature vector</u>

Scatter plot of fish width vs. lightness





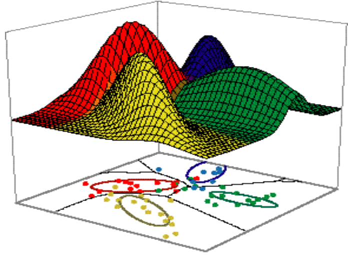


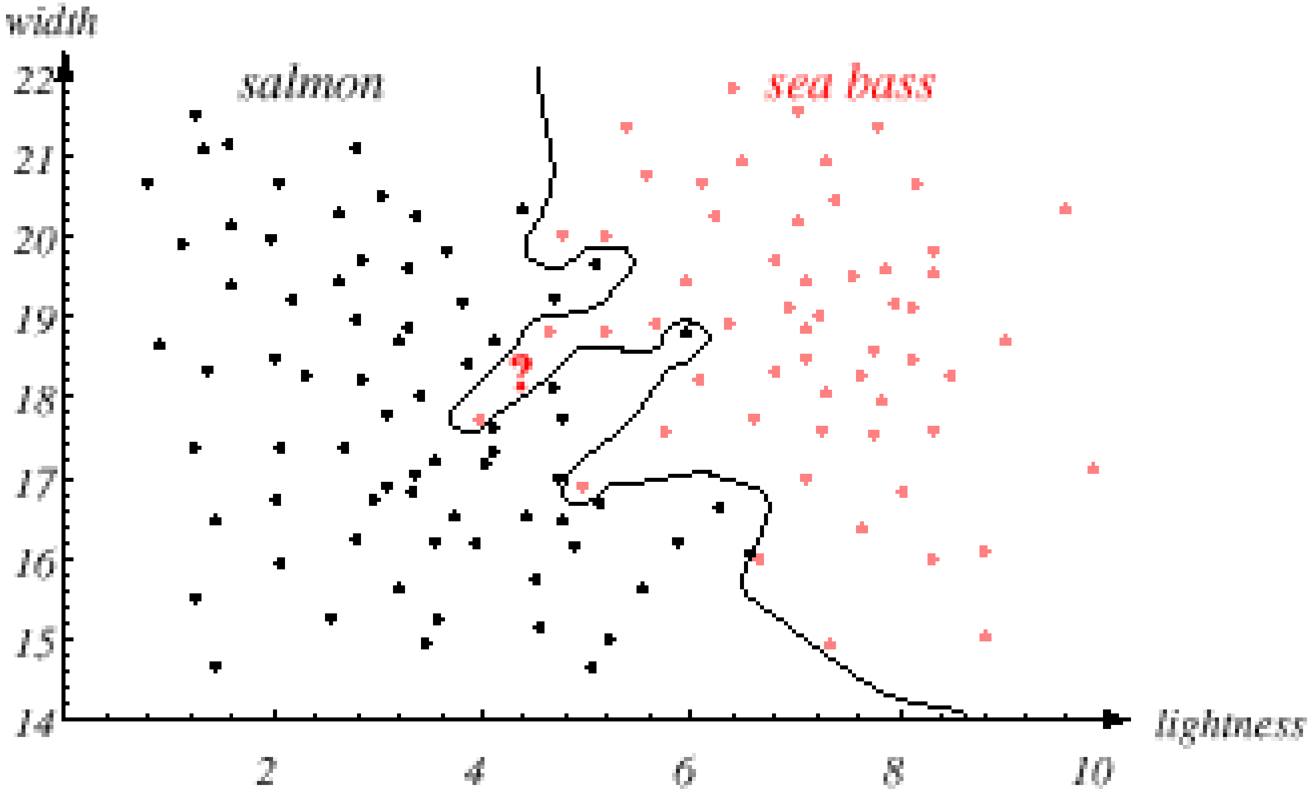
Make the classifier more complex?

- 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 “noisy features”

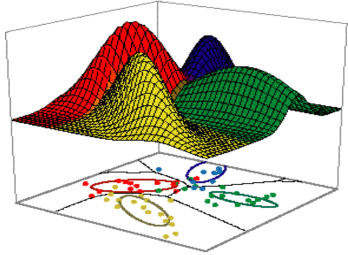
- We are tempted to believe that the best decision boundary should be the one which provides an optimal performance such as in the following figure:

An ‘optimal’ decision boundary?





Overfitting and generalization

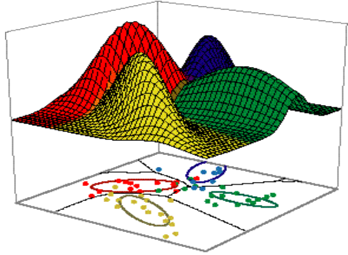


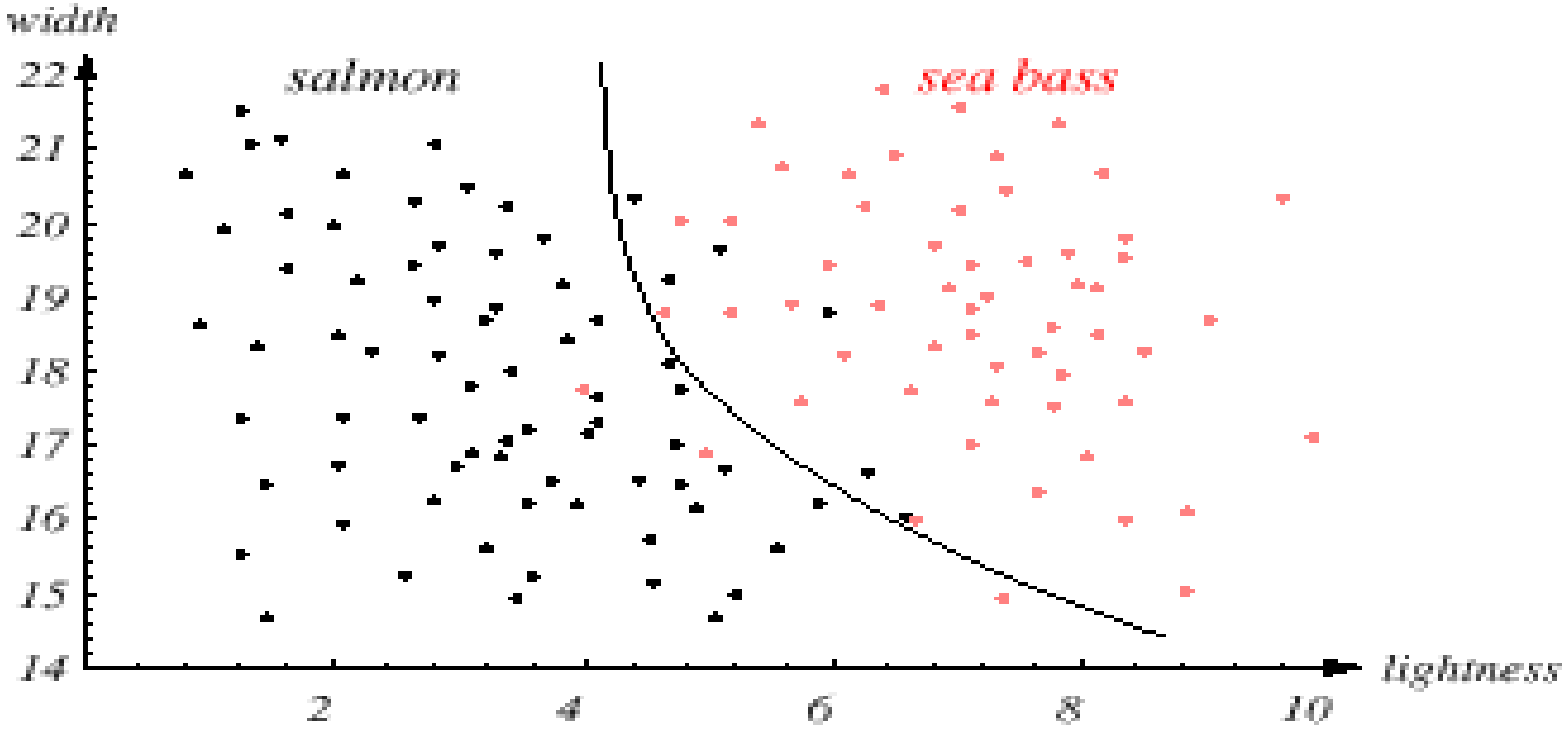
- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel input, not just training example inputs.

Issue of generalization!

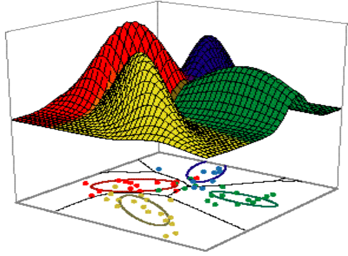
- Performance on the training data is not always indicative of performance on future test data

An improved decision boundary

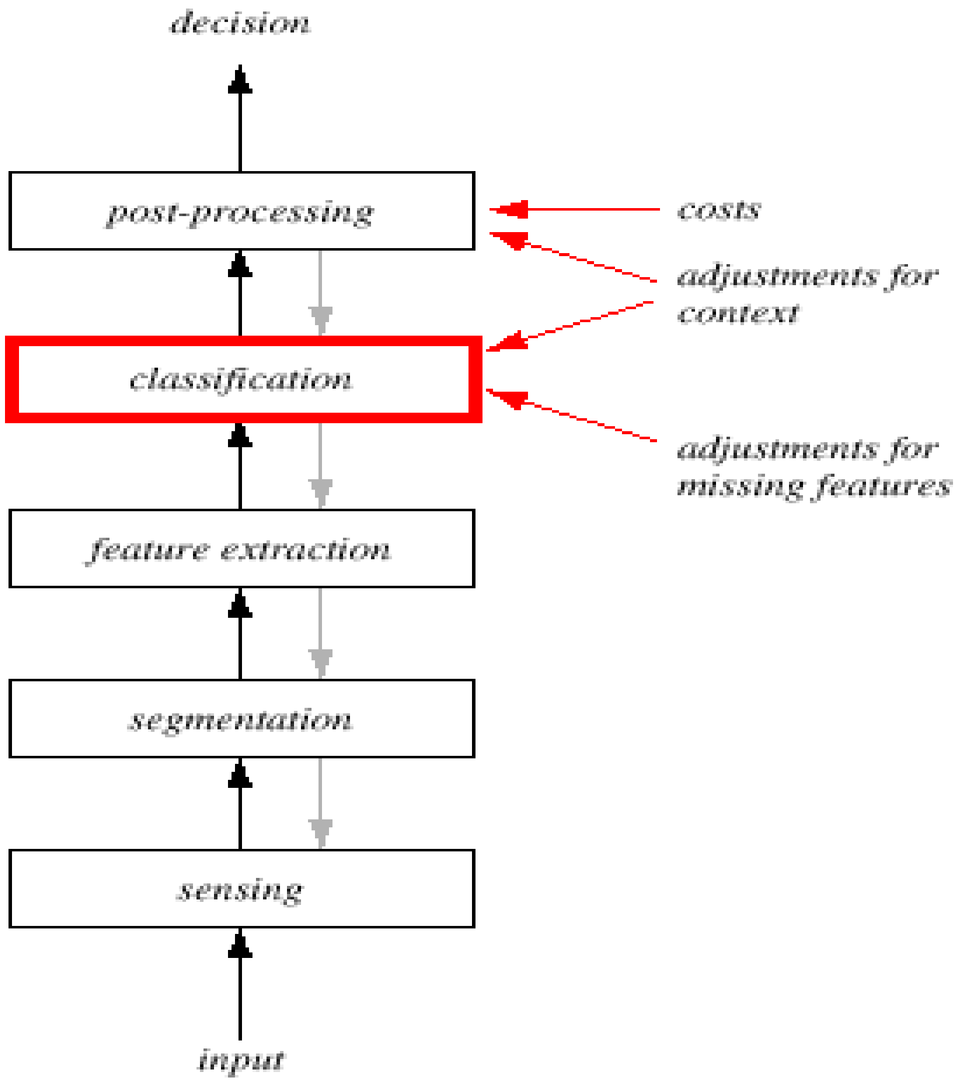




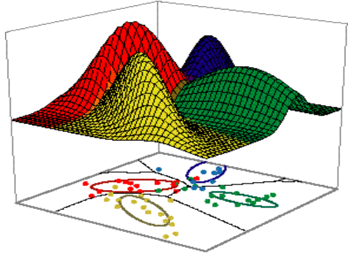
Pattern Recognition Systems



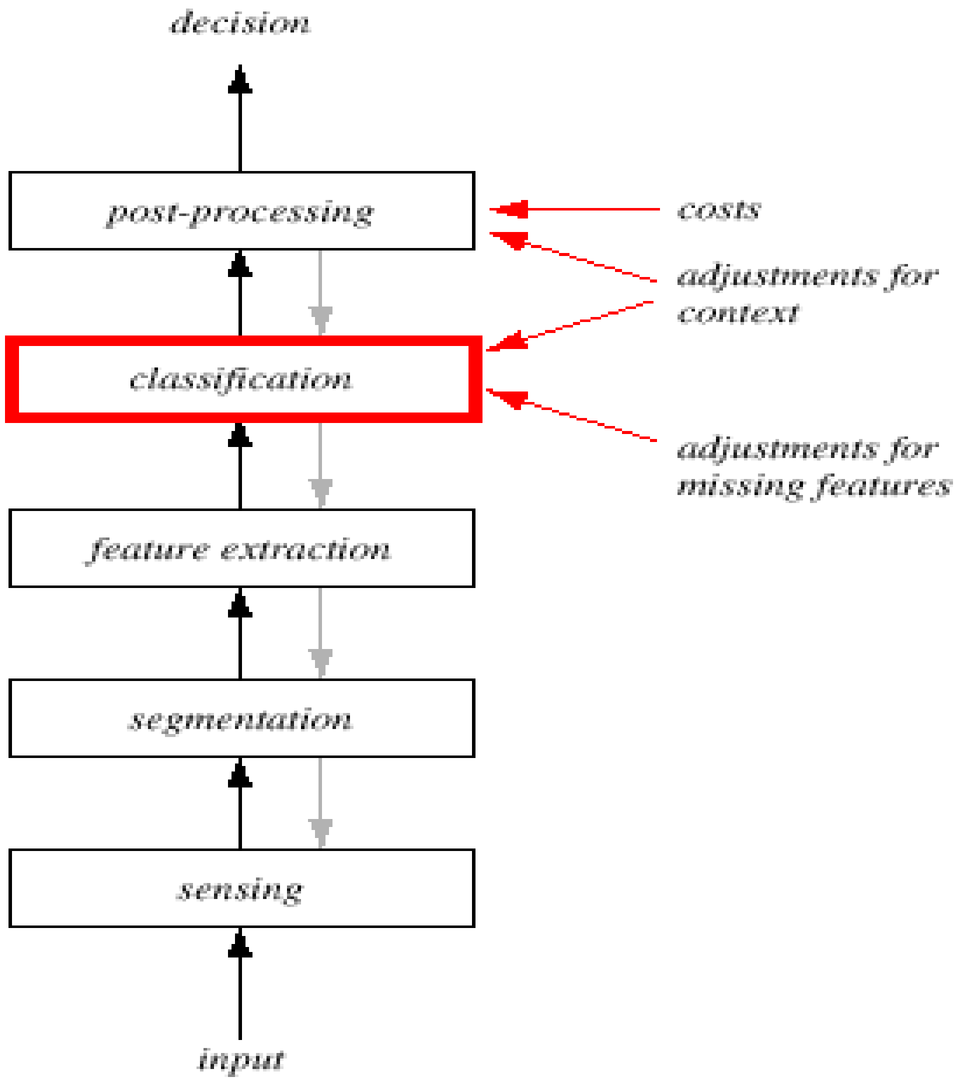
- Sensing
  - Use of a transducer (e.g. camera or microphone)
  - PR system depends of the bandwidth, the resolution, sensitivity, distortion, and noise of the transducer



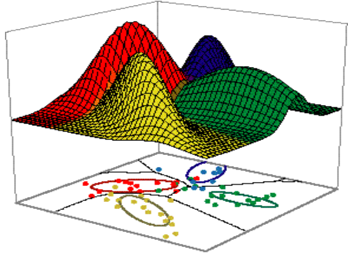
Pattern Recognition Systems



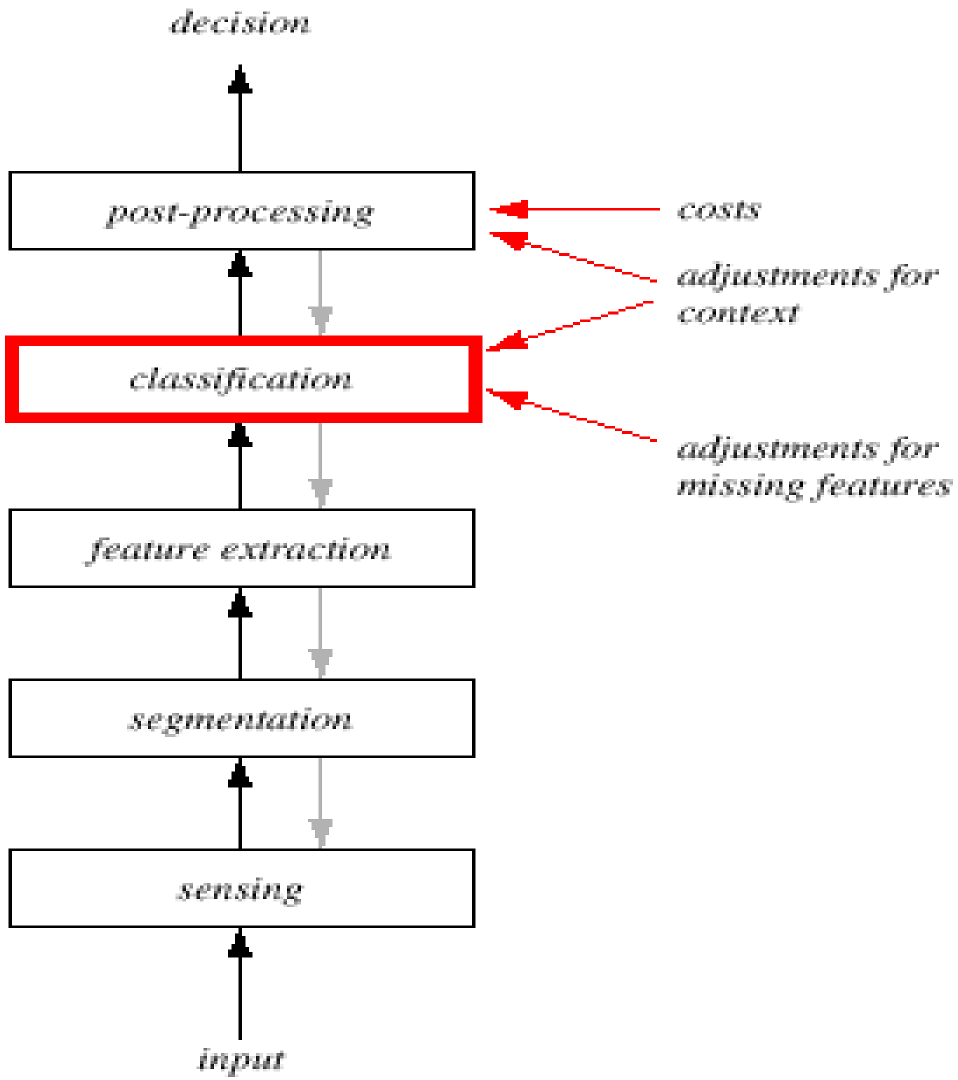
- Segmentation
  - Extract object of interest from background.
  - Patterns should be well separated and should not overlap



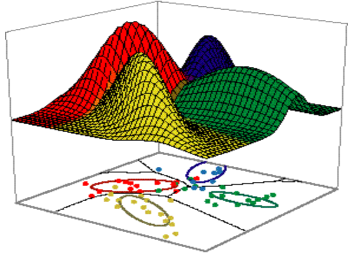
Pattern Recognition Systems



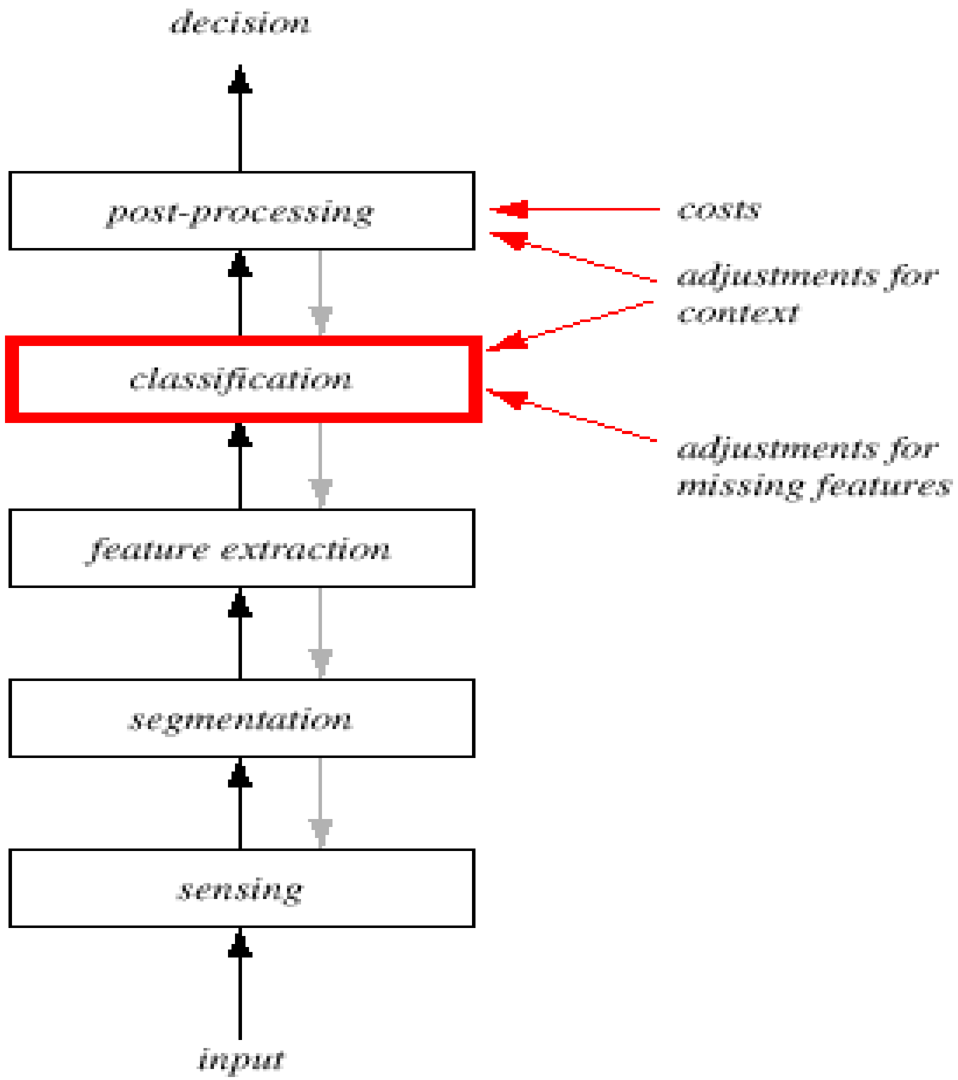
- Feature extraction
  - Discriminative features
  - May require invariant features with respect to translation, rotation and scale.
  - If features are powerful, then classifier can be trivial.



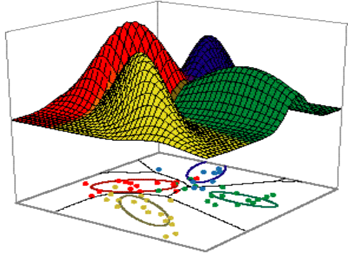
Pattern Recognition Systems



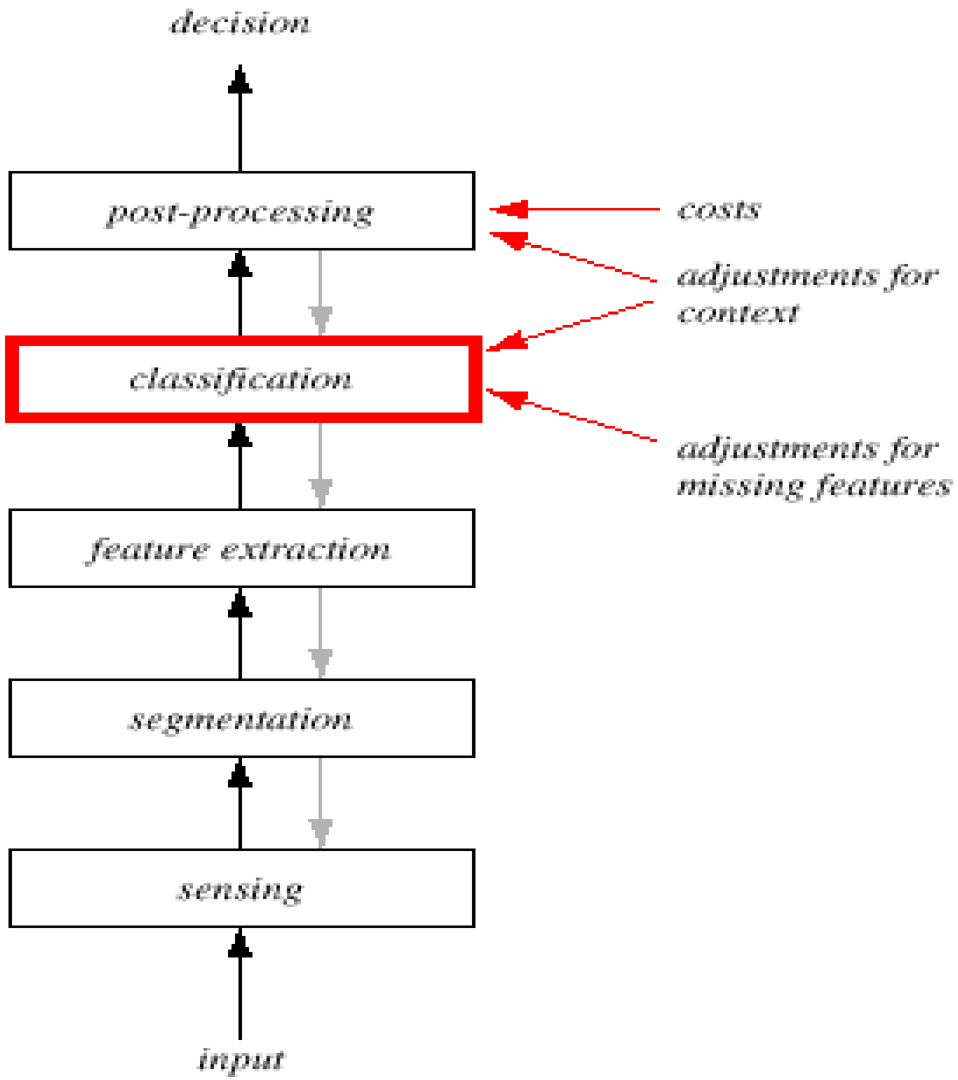
- Classification
  - Use a feature vector provided by a feature extractor to assign the object to a category



Pattern Recognition Systems

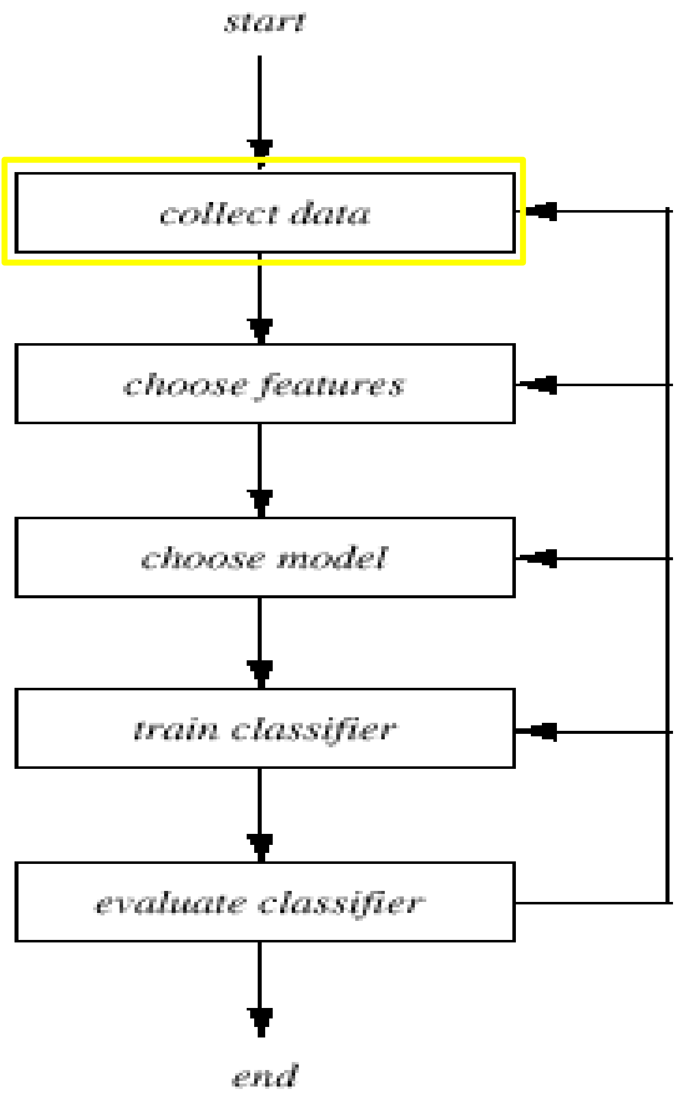


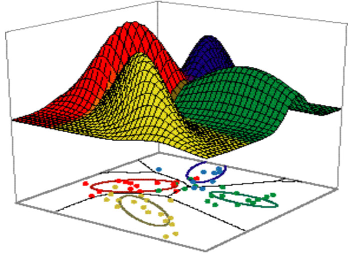
- Post Processing
  - Exploit context dependent information other than from the target pattern itself to improve performance
  - May also adjust for costs
  - Can combine multiple classifiers



The Design Cycle

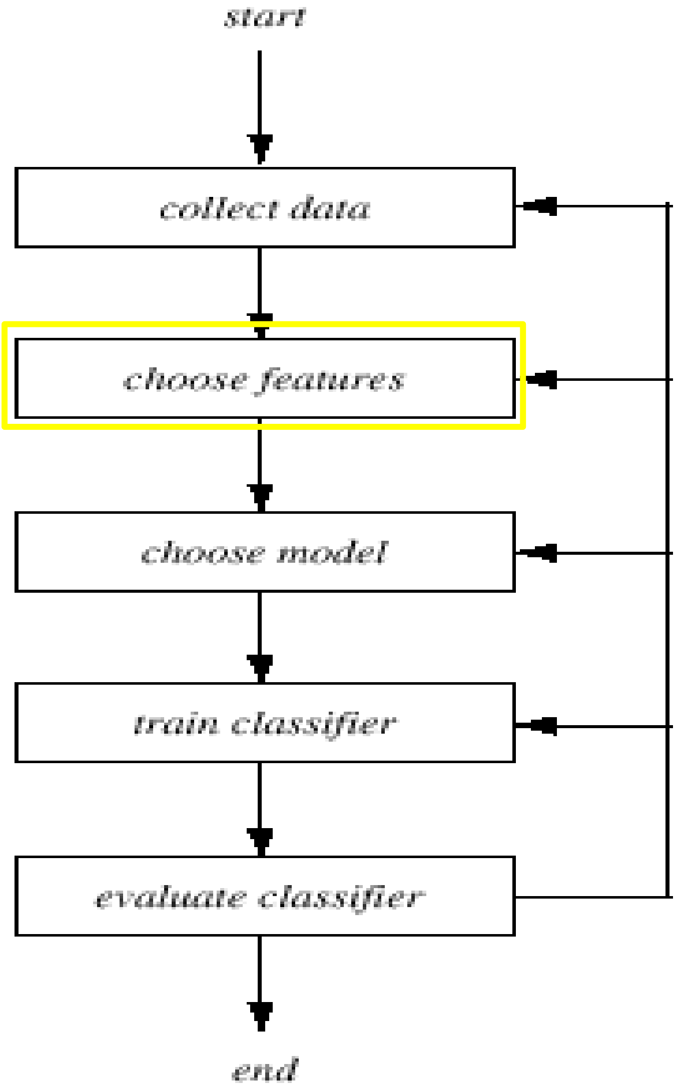
- Data Collection
  - How do we know when we have collected an adequately large and representative set of examples for training and testing the system?



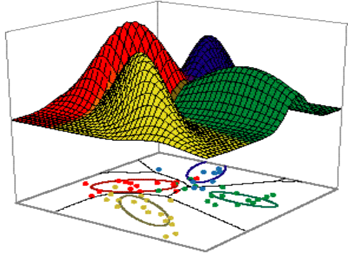


The Design Cycle

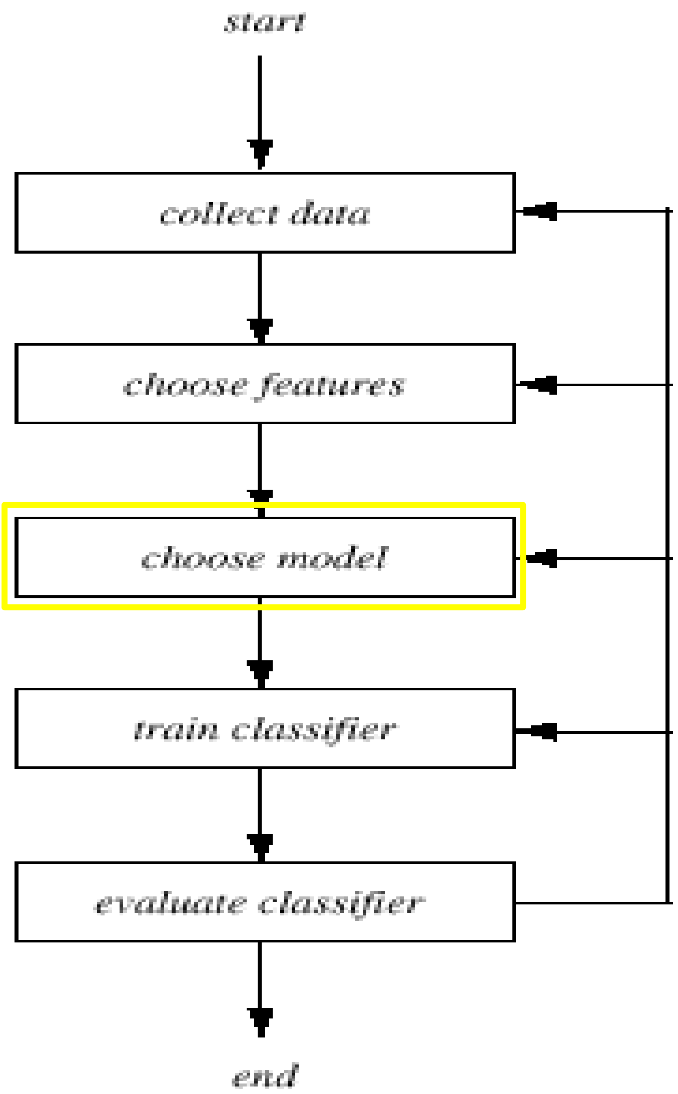
- Feature Choice
  - Depends on the characteristics of the problem domain. Simple to extract, invariant to irrelevant transformation insensitive to noise.
  - Use prior information about the problem.
    - E.g. fishermen tell you salmon are shorter.



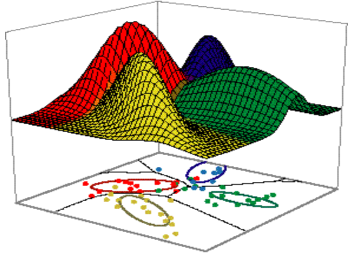
The Design Cycle



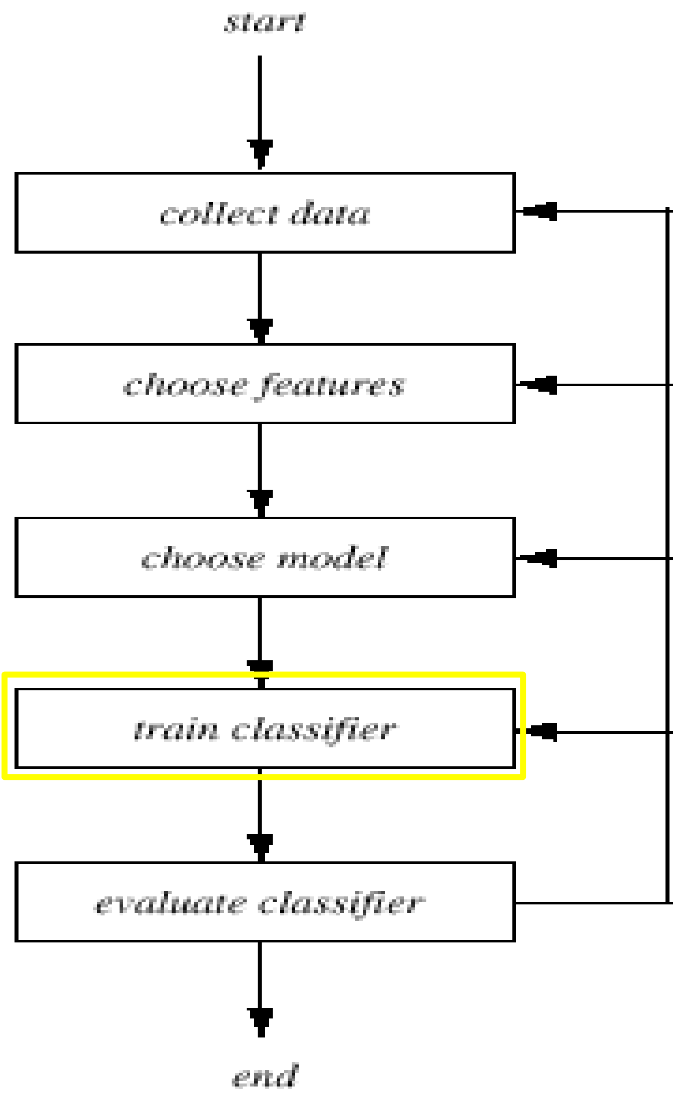
- Model Choice
  - Model is our representation of a fish (width and lightness)
  - If unsatisfied with the performance of our fish classifier and want to jump to another class of model (different features)
  - Use prior information about the problem.



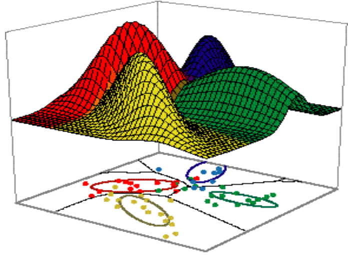
The Design Cycle

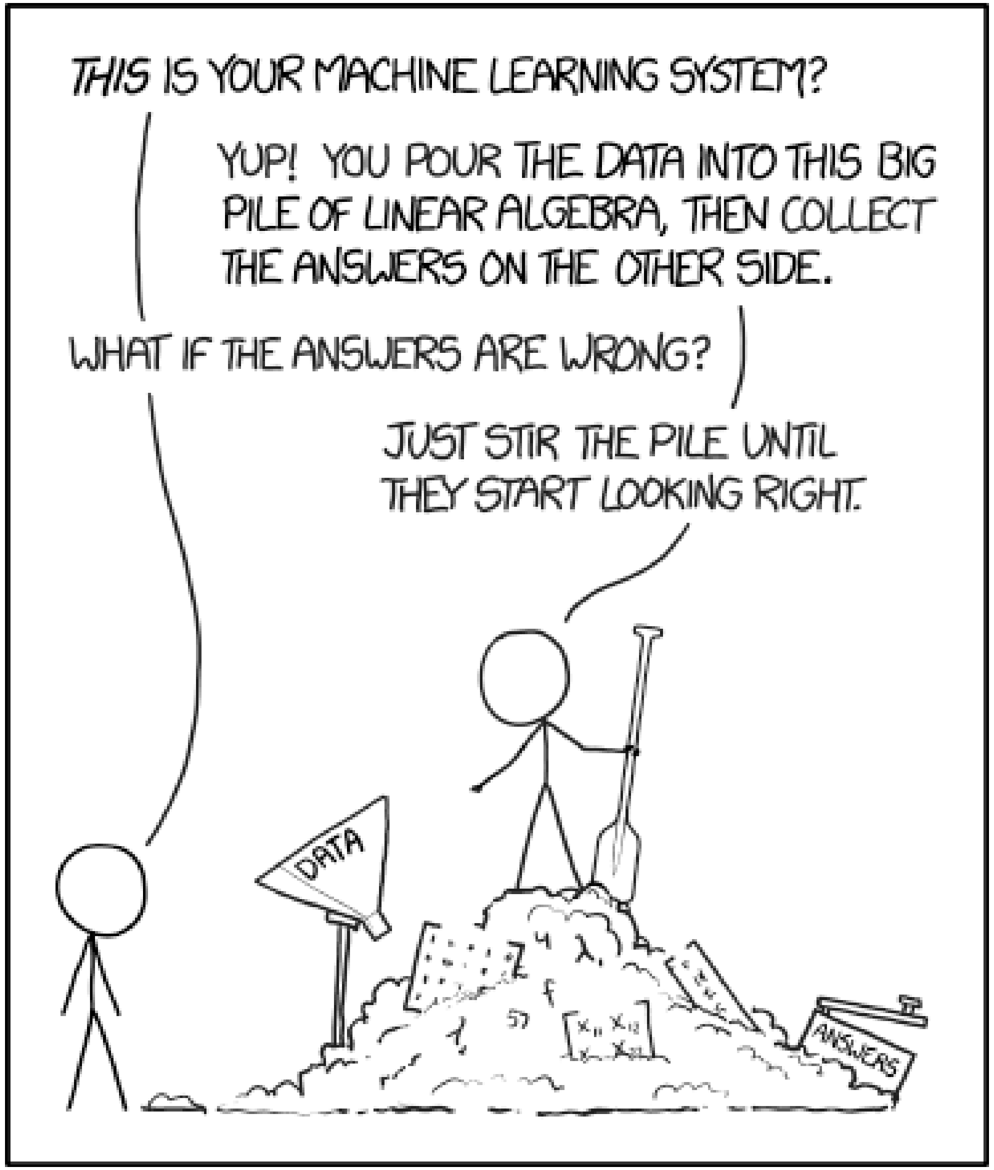


- Training
  - Use data to determine the classifier. Many different procedures for training classifiers and choosing models



Training = “Just keep stirring”

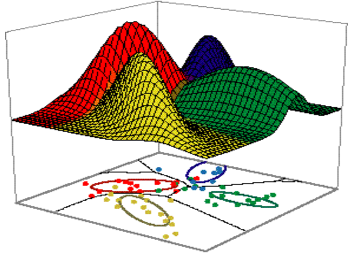


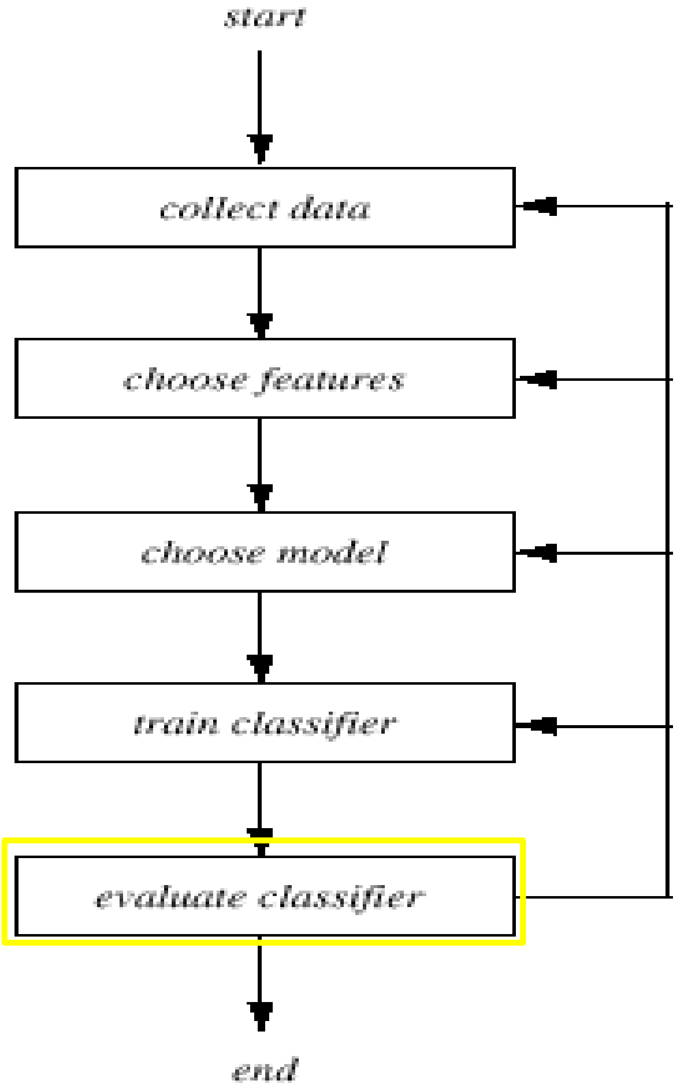


https://xkcd.com/1838/

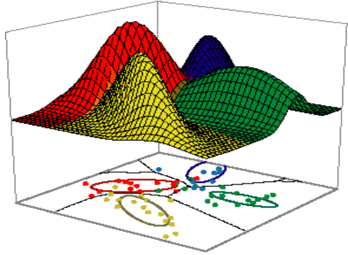
The Design Cycle

- Evaluation
  - Measure the error rate (or performance) and switch model, features, classifier, training algorithm, etc.



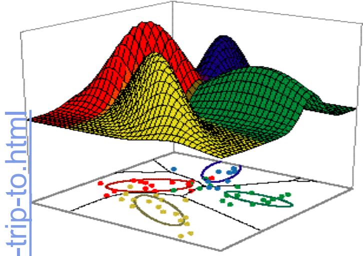


Computational Complexity



- 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…
  - Not the focus of this course.

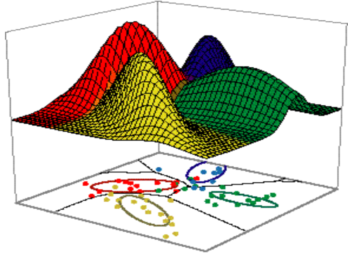
Unsupervised Learning



- Cluster these items:

http://www.lifeofcharmings.com/2012/11/obsessed-random-favs-from-our-trip-to.html

Learning and Adaptation

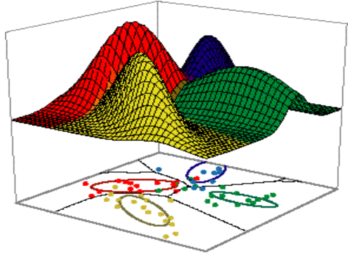


- Supervised learning
  - A teacher provides a category label or cost for each pattern in the training set
  - Classifier is given features (inputs) and also answers (outputs)
  - Comparing the input and the output lets the training algorithm see what it needs to learn
  - Task is to learn a function converting inputs to their corresponding outputs

- Unsupervised learning
  - Learner receives only input features, but no classifiction output values
  - During training, classifier is not told "what to do"
  - It looks at the data and tries to find patterns
    - Useful for figuring out what types of inputs are likely to occur
  - The system forms clusters or "natural groupings" of the input patterns

- Semi-supervised learning
  - Learn from both labelled and unlabelled data

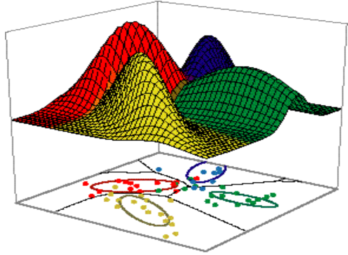
Learning and Adaptation



- Parametric models
  - Feature data for each class can be described by some parameterized distribution
  - Decision boundary a function of those parameters
  - Goal is to estimate the parameters correctly

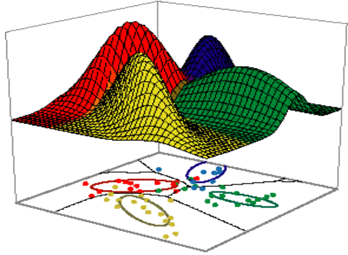
- Nonparametric
  - No assumptions made about the data
  - Decision boundary directly depends on data
  - Goal is to determine decision boundary directly

Related Fields



- Image Processing
- Associative Memory
- Regression
- Interpolation
- Density Estimation
- Data Mining
- Machine Learning
- Deep Learning
- Natural Language Processing
- Artificial Intelligence

Course Goals



- Be familiar with a variety of pattern classification approaches
  - Statistical
  - Machine learning
  - Nonparametric
  - Unsupervised
- Understand how to set up your experiment such that you will obtain valid conclusions from your results
  - Construction of datasets & partitioning
  - Hypothesis testing and statistical tests
- Be able to critically assess the claims of others