Introduction to Pattern Classification

NASA Space Program

1

Outline

- Classification and its Applications
- An Example
- Pattern Recognition Systems
- The Design Cycle
- Learning and Adaptation
- Conclusion

2

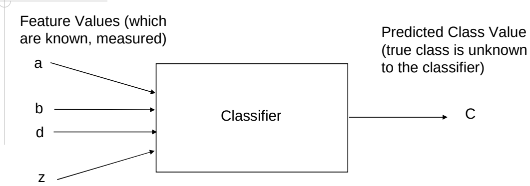
Classification

- Classification is an important component of intelligent systems
- We have a special discrete-valued variable called the Class, C takes values in $\{c_1, c_2, \ldots\ c_m\}$
- Problem is to decide what class an object is
  - i.e., what value the class variable C is for a given object
  - given measurements on the object, e.g., A, B, ...
  - These measurements are called "features"

  we wish to learn a mapping from Features -> Class
- Notation:
  - C is the class
  - A, B, etc (the measurements) are called the "features" (sometimes also called "attributes")

3

Classification Functions



We want a mapping or function which takes any combination of values $\underline{x}$ = (a, b, d, ... z) and will produce a prediction C, i.e., a function C = f(a, b, d, …. z) which produces a value c_1, c_2, …. or c_m

The problem is that we don't know this mapping: we have to learn it from data!

4

Applications of Classification

- Medical Diagnosis
  - classification of cancerous cells
- Credit card and Loan approval
  - Most major banks
- Speech recognition
  - IBM, Dragon Systems, AT&T, Microsoft, etc
- Optical Character/Handwriting Recognition
  - Post Offices, Banks, Gateway, Motorola, Microsoft, Xerox, etc.
- Email classification
  - classify email as "junk" or "non-junk"
- Many other applications
  - one of the most successful applications of AI technology

5

Examples of Features and Classes

Application	Class Names	Features
Cancer Cell Diagnosis		
Zipcode Recognition		
Credit Card Approval		
Speech Recognition		
Email Classification		

6

Examples of Features and Classes

Application	Class Names	Features
Cancer Cell Diagnosis	Cancer, No Cancer	
Zipcode Recognition	1,2,3,.....9,0, "other"	
Credit Card Approval	Good risk, bad risk	
Speech Recognition	Vocabulary of words (e.g., 5000)	
Email Classification	Junk, non-junk	

7

Examples of Features and Classes

Application	Class Names	Features
Cancer Cell Diagnosis	Cancer, No Cancer	Estimated Shape, Size, Number, Color of Cells
Zipcode Recognition	1,2,3,.....9,0, "other"	Relative pixel locations in image, angles/lengths of lines in image
Credit Card Approval	Good risk, bad risk	Income, current debts, age, education level, etc.
Speech Recognition	Vocabulary of words (e.g., 5000)	Features extracted from spectrum of speech signal
Email Classification	Junk, non-junk	Words in subject and text

8

Feature Vectors and Feature Spaces

- Feature Vector:
  - Say we have 2 features: we can think of the features as a 2-component vector
    - i.e., a 2-dimensional vector, [a b]
  - So the features correspond to a 2-dimensional space
    - (clearly we can generalize to d-dimensional space)
    - this is called the "feature space"
  - Each feature vector represents the "coordinates" of a particular object in feature space
  - If the feature-space is 2-dimensional (for example), and the features a and b are real-valued
    - we can visually examine and plot the locations of the feature vectors

9

Decision Boundaries

- What is a Classifier?
  - A classifier is a *mapping* from feature space (a d-dimensional vector) to the class labels {1, 2, … m}
  - Thus, a classifier partitions the feature space into m *decision regions*
  - The line or surface separating any 2 classes is the *decision boundary*
- Linear Classifiers
  - a linear classifier is a mapping which partitions feature space using a linear function (a straight line, or a hyperplane)
  - it is one of the simplest classifiers we can imagine
  - in 2 dimensions the decision boundary is a straight line

10

Class Overlap

- Consider two class case
  - data from D1 and D2 may **overlap**
    - features = {age, body temperature}, classes = {flu, not-flu}
    - features = {income, savings}, classes = {good/bad risk}
  - common in practice that the classes will naturally overlap
    - this means that our features are usually not able to perfectly discriminate between the classes
    - note: with more expensive/more detailed additional features (e.g., a specific test for the flu) we might be able to get perfect separation
  - if there is overlap => classes are not linearly separable

11

Data from Multiple Classes

- Now consider that we have data from m classes (e.g., m=2)
- We can imagine the data from each class being in a "cloud" in feature space
  - data sets D1 and D2 (sets of points from classes 1 and 2)
  - data are of dimension d (i.e., d-dimensional vectors)
  - if d = 2 (2 features), we can plot the data
    - we should see two "clouds" of data points, one cloud per class

12

An Example

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

Species → Sea bass

Species → Salmon

13

An Example

- 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…

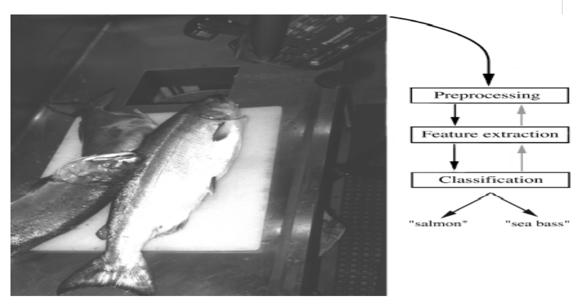
    - This is the set of all suggested features to explore for use in our classifier!

14

An Example

- 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

15



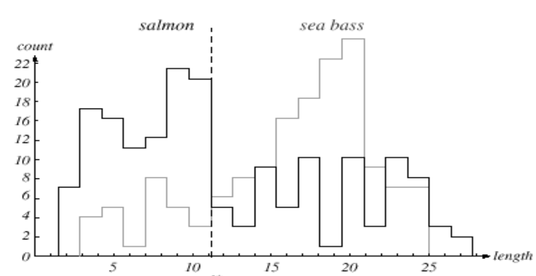
16

An Example

- Classification

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

17



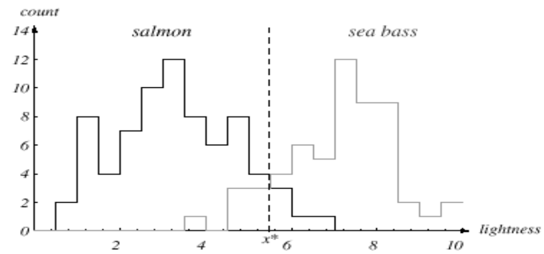
18

An Example

The length is a poor feature alone!

Select the lightness as a possible feature.

19



20

An Example

- 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

21

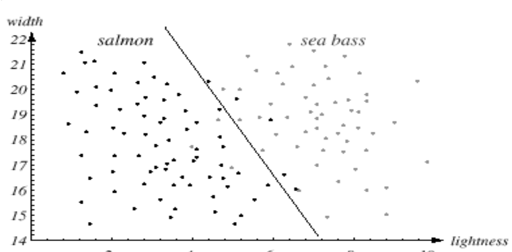
An Example

- Adopt the lightness and add the width of the fish

Fish ⇒ $x^T = [x_1, x_2]$

x_1 → **Lightness**, x_2 → **Width**

22

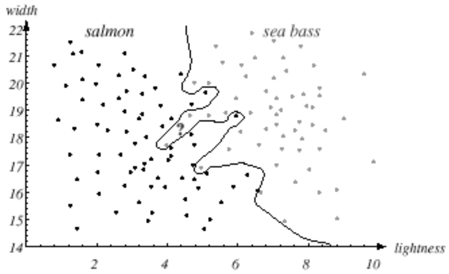


23

An Example

- 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:

24



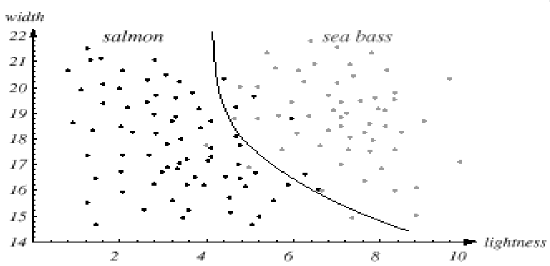
25

An Example

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

Issue of generalization!

26



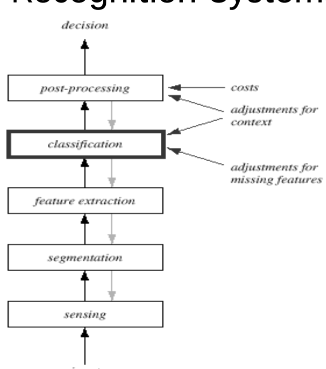
27

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

28

Pattern Recognition Systems



29

Pattern Recognition Systems

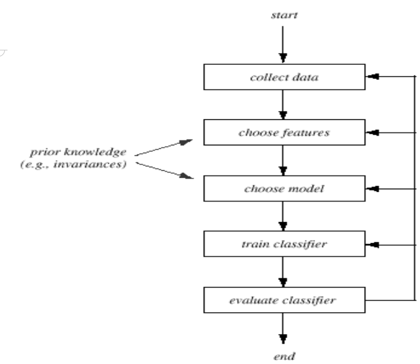
- Feature extraction
  - Discriminative features
  - Invariant features with respect to translation, rotation and scale
- Classification
  - Use a feature vector provided by a feature extractor to assign the object to a category
- Post Processing
  - Exploit context input dependent information other than from the target pattern itself to improve performance

30

The Design Cycle

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

31



32

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

33

The Design Cycle

- 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

34

The Design Cycle

- 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?)

35

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

36

Conclusion

- Reader seems to be overwhelmed by the number, 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

37