BIM488 Introduction to Pattern Recognition

Introduction

Outline

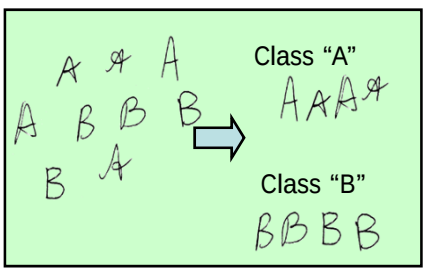
- Pattern Recognition
- An Example
- Pattern Recognition Systems
- The Design Cycle

Pattern Recognition

- A **pattern**, from the French *patron*, is a type of theme of recurring events or objects, sometimes referred to as elements of a set of objects.
- Arrangement of objects which has a mathematical, geometric, statistical etc. relationship
- A pattern is an abstract object, such as a set of measurements describing a physical object.

Pattern Recognition

- Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes.



Pattern Recognition

- Depending on the application, these objects can be
  - images
  - signal waveforms
  - text
  - or any type of measurements

  that need to be classified.
- We will refer to these objects using the generic term **patterns**.
- The task is to assign unknown patterns into the correct class which is known as classification.

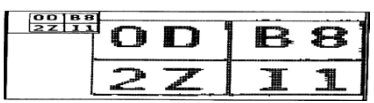
Pattern Recognition

Pattern Class

- A collection of “similar” (not necessarily identical) objects
  - Intra-class variability

The letter “T” in different typefaces

  - Inter-class variability



Characters that look similar

Pattern Recognition

- Pattern recognition applications

Problem Domain	Application	Input Pattern	Pattern Classes
Document image analysis	Optical character recognition	Document image	Characters, words
Document classification	Internet search	Text document	Semantic categories
Document classification	Junk mail filtering	Email	Junk/non-junk
Multimedia database retrieval	Internet search	Video clip	Video genres
Speech recognition	Telephone directory assistance	Speech waveform	Spoken words
Natural language processing	Information extraction	Sentences	Parts of speech
Biometric recognition	Personal identification	Face, iris, fingerprint	Authorized users for access control
Medical	Diagnosis	Microscopic image	Cancerous/healthy cell
Military	Automatic target recognition	Optical or infrared image	Target type
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective/non-defective product
Industrial automation	Fruit sorting	Images taken on a conveyor belt	Grade of quality
Remote sensing	Forecasting crop yield	Multispectral image	Land use categories
Bioinformatics	Sequence analysis	DNA sequence	Known types of genes
Data mining	Searching for meaningful patterns	Points in multidimensional space	Compact and well-separated clusters

Pattern Recognition

Main PR Methods:

- *Statistical pattern recognition (**we will focus on this**)*
  - Focuses on the statistical properties of the patterns (i.e., probability densities).
- *Structural pattern recognition*
  - Describe complicated objects in terms of simple primitives and structural relationships.
- *Syntactic pattern recognition*
  - Decisions consist of logical rules or grammars.
- *Template matching*
  - The pattern to be recognized is matched against a stored template while taking into account all allowable pose (translation and rotation) and scale changes.

An Example

- Problem: Sorting incoming fish on a conveyor belt according to species
- Assume that we have only two kinds of fish:
  - sea bass
  - salmon

An Example: Decision Process

- What kind of information can distinguish one species from the other?
  - Length
  - Width
  - Weight
  - Number and shape of fins
  - Tail shape
  - etc.

An Example: Selecting Features

- Assume a fisherman told us that a sea bass is generally longer than a salmon.
- We can use length as a feature and decide between sea bass and salmon according to a threshold on length.
- But, how can we choose this threshold?

An Example: Selecting Features

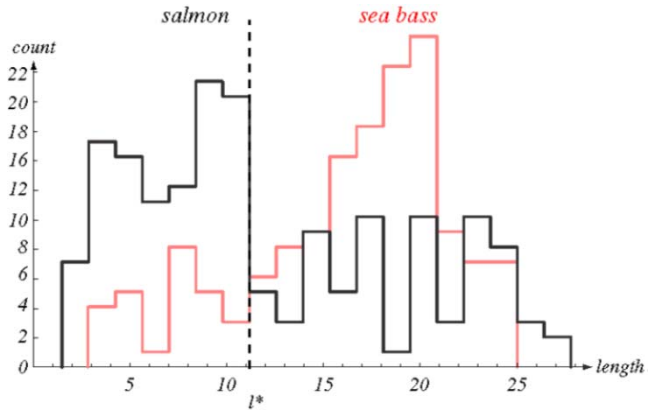


Figure: Histograms of the length feature for two types of fish in training samples.

- How can we choose the threshold l* to make a reliable decision?

An Example: Selecting Features

- In statistics, a **histogram** is a graphical representation showing a visual impression of the distribution of data.
- It is an estimate of the probability distribution of a variable.

An Example: Selecting Features

- Even though sea bass is longer than salmon on the average, there are many examples of fish where this observation does not hold.

- Try another feature: average lightness of the fish scales.

An Example: Selecting Features

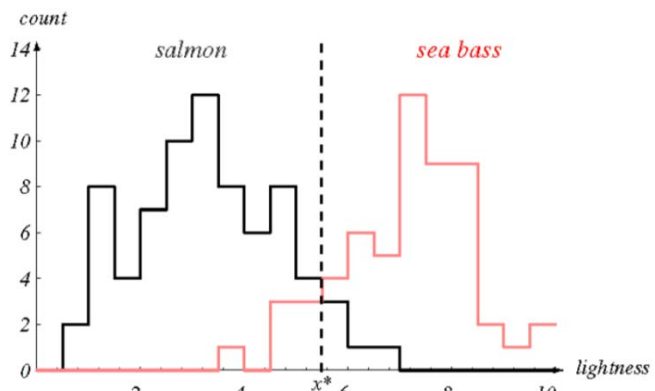


Figure: Histograms of the lightness feature for two types of fish in training samples.

- How can we choose the threshold x* to make a reliable decision?

An Example: Cost of Error

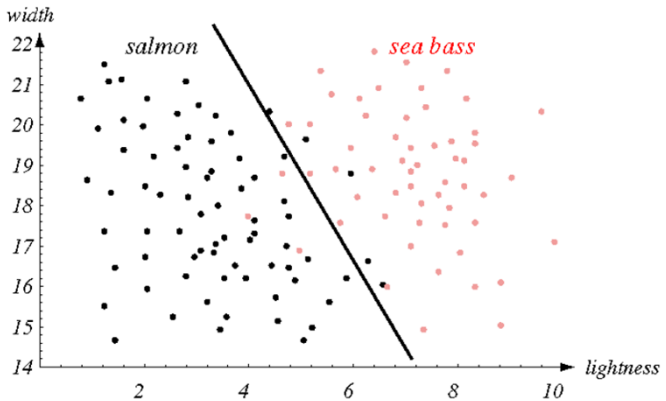
- We should also consider costs of different errors we make in our decisions.
- For example, if the fish packing company knows that:
  - Customers who buy salmon will be angry if they see cheap sea bass in their cans.
  - Customers who buy sea bass will not be unhappy if they occasionally see some expensive salmon in their cans.
- How does this knowledge affect our decision?

An Example: Multiple Features

- Assume we also observed that sea bass are typically wider than salmon.
- We can use two features in our decision:
  - lightness: x1
  - width: x2
- Each fish image is now represented as a point (**feature vector**) in a two-dimensional feature space:

$$x = [x1\ x2]$$

An Example: Multiple Features



- Figure: Scatter plot of lightness and width features for training samples. We can draw a decision boundary to divide the feature space into two regions. Does it look better than using only lightness?

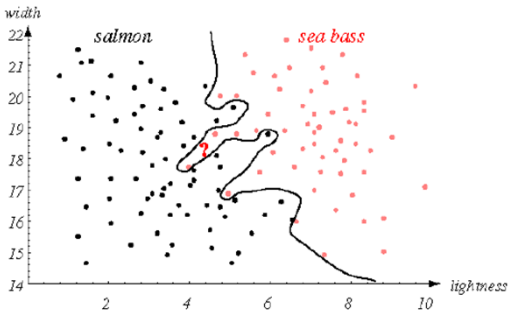
An Example: Multiple Features

- Does adding more features always improve the results?
  - Avoid unreliable features.
  - Be careful about correlations with existing features.
  - Be careful about measurement costs.
  - Be careful about noise in the measurements.

- Is there some curse for working in very high dimensions?

An Example: Decision Boundaries

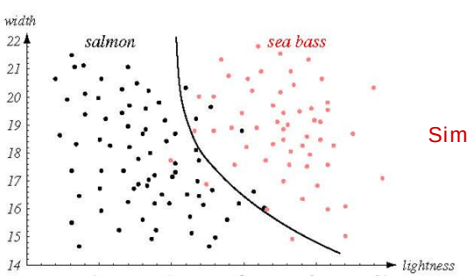
- Can we do better with another decision rule?
- More complex models result in more complex boundaries.



- Figure: We may distinguish training samples perfectly but how can we predict how well we can generalize to unknown samples?

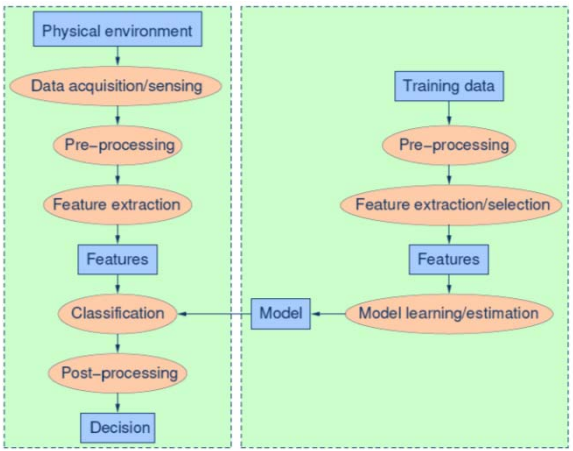
An Example: Generalization

- The ability of the classifier to produce correct results on *novel* patterns.
- How can we improve generalization performance ?
  - More training examples (i.e., better pdf estimates).
  - Simpler models (i.e., simpler classification boundaries) usually yield better performance.



Simplify the decision boundary!

Pattern Recognition Systems



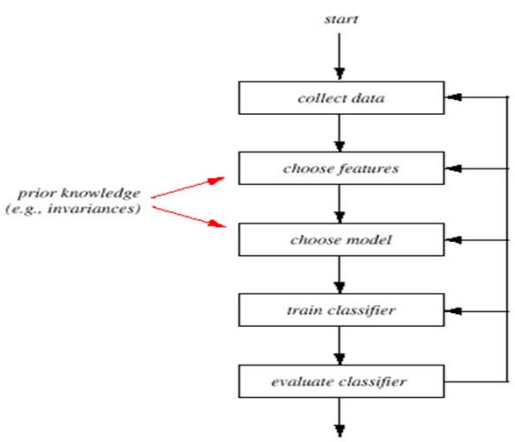
Pattern Recognition Systems

- Data acquisition and sensing:
  - Measurements of physical variables
  - Important issues: bandwidth, resolution, sensitivity, distortion, SNR, latency, etc.
- Pre-processing:
  - Removal of noise in data
  - Isolation of patterns of interest from the background
- Feature extraction:
  - Finding a new representation in terms of features

Pattern Recognition Systems

- Model learning and estimation:
  - Learning a mapping between features and pattern groups and categories
- Classification:
  - Using features and learned models to assign a pattern to a category
- Post-processing:
  - Evaluation of confidence in decisions
  - Exploitation of context to improve performance
  - Combination of experts

The Design Cycle



The Design Cycle: Overview of Important Issues

- Noise
- Data Collection / Feature Extraction
- Pattern Representation / Invariance/Missing Features
- Model Selection / Overfitting
- Prior Knowledge / Context
- Classifier Combination
- Costs and Risks
- Computational Complexity

The Design Cycle: Issue: Noise

- Various types of noise (e.g., shadows, conveyor belt might shake, etc.)
- Noise can reduce the reliability of the feature values measured.
- Knowledge of the noise process can help to improve performance.

The Design Cycle: Issue: Data Collection

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

The Design Cycle: Issue: Feature Extraction

- It is a domain-specific problem which influences classifier's performance.
- Which features are most promising ?
- Are there ways to automatically learn which features are best ?
- How many should we use ?
- Choose features that are robust to noise.
- Favor features that lead to simpler decision regions.

The Design Cycle: Issue: Pattern Representation

- Similar patterns should have similar representations.
- Patterns from different classes should have dissimilar representations.
- Pattern representations should be invariant to transformations such as:
  - translations, rotations, size, reflections, non-rigid deformations
- Small intra-class variation, large inter-class variation.

The Design Cycle: Issue: Missing Features

- Certain features might be missing (e.g., due to occlusion).
- How should the classifier make the best decision with missing features ?
- How should we train the classifier with missing features ?

The Design Cycle: Issue: Model Selection

- How do we know when to reject a class of models and try another one ?
- Is the model selection process just a trial and error process ?
- Can we automate this process ?

The Design Cycle: Issue: Overfitting

- Models complex than necessary lead to *overfitting* (i.e., good performance on the training data but poor performance on novel data).
- How can we adjust the complexity of the model ? (not very complex or simple).
- Are there principled methods for finding the best complexity ?

The Design Cycle: Issue: Context

How m ch
info mation are
y u mi sing?

The Design Cycle: Issue: Classifier Combination

- Performance can be improved using a "pool" of classifiers.
- How should we combine multiple classifiers ?

The Design Cycle: Issue: Costs and Risks

- Each classification is associated with a cost or risk (e.g., classification error).
- How can we incorporate knowledge about such risks ?
- Can we estimate the *lowest* possible risk of *any* classifier ?

The Design Cycle: Issue: Computational Complexity

- How does an algorithm *scale* with
  - the number of feature dimensions
  - number of patterns
  - number of categories
- Brute-force approaches might lead to perfect classifications results but usually have impractical time and memory requirements.
- What is the tradeoff between computational ease and performance ?

The Design Cycle: General Purpose PR Systems?

- Humans have the ability to switch rapidly and seamlessly between different pattern recognition tasks
- It is very difficult to design a device that is capable of performing a variety of classification tasks
  - Different decision tasks may require different features.
  - Different features might yield different solutions.
  - Different tradeoffs (e.g., classification error vs processing time) exist for different tasks.

Summary

- Pattern Recognition
- An Example
- Pattern Recognition Systems
- The Design Cycle

References

- S. Theodoridis and K. Koutroumbas, *Pattern Recognition* (4th Edition), Academic Press, 2009.

- R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification* (2nd Edition), Wiley, 2001.