Introduction to Machine Learning
Reykjavík University
Spring 2007

Instructor: Dan Lizotte

Logistics

- To contact Dan:
  - dlizotte@cs.ualberta.ca
  - http://www.cs.ualberta.ca/~dlizotte/teaching/
- Books:
  - Introduction to Machine Learning, Alpaydin
    - We'll use mostly this one
  - Reinforcement Learning: An Introduction
    - We'll use this somewhat at the end - it's online

Logistics

- Time
  - MTWRF, 8:15am - 9:00am, 9:15am - 10:00am
- Lectures
  - K21 (Kringlan 1)
- Labs
  - Room 432 (Ofanleiti 2)

What is Machine Learning

- “Machine learning is programming computers to optimize a performance criterion using example data or past experience.”
  - Alpaydin
- “The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.”
  - Mitchell
- “…the subfield of AI concerned with programs that learn from experience.”
  - Russell & Norvig

What else is Machine Learning?

- Data Mining
  - "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data."
    - W. Frawley and G. Piatetsky-Shapiro and C. Matheus
  - "..the science of extracting useful information from large data sets or databases."
    - D. Hand, H. Mannila, P. Smyth
  - "Data-driven discovery of models and patterns from massive observational data sets."
    - Padhraic Smyth

This is all pretty vague…

- You may find that in this course, we cover a bunch of loosely related topics.
  - You're right.
  - That's kind of what ML is.
- Hopefully, you will learn a little bit about a lot of things
  - Some theory
  - Some practice
- To get the most out of this course,
  - ASK ME QUESTIONS

Any questions before we start?

- Anybody?
  - Anybody?
    - Really people -- now is the time…

- …but you can (and should) always ask later.
- Let's look at a few examples.
  - Alpaydin, Ch 1.2

Learning Associations

- What things go together?
  - Chips and beer, maybe?
- Suppose we want P(chips|beer). "The probability a particular customer will buy chips, given that he or she has bought beer."
- We will estimate this probability from data.
- P(chips|beer) ≈ #(chips & beer) / #beer
- Just count the people who bought beer ***and*** chips, and divide by the number of people who bought beer
- While not glamorous, counting is learning.

Classification

- Input: “features” Output: “label”
  - Features can be symbols, real numbers, etc…
    - [age, height, weight, gender, hair_colour, …]
  - Labels come from a (small) discrete set
    - L = {Icelander, Canadian}
- We need a *discriminant* function that maps feature vectors to labels.
- We can learn this from data, in many ways.
  - ([27, 172, 68, M, brown, …], Canadian)
  - ([29, 160, 54, F, brown, …], Icelander)
  - …
- We can use it to *predict* the label of a new instance.
  - How good are our predictions?

Regression

- Input: “features” Output: “response”
  - Features can be symbols, real numbers, etc…
    - [age, height, weight, gender, hair_colour, …]
  - Response is *real-valued.*
    - $-\infty < life\_span < \infty$
- We need a *regression* function that maps feature vectors to responses.
- We can learn this from data, in many ways.
  - ([27, 172, 68, M, brown, …], 86)
  - ([29, 160, 54, F, brown, …], 99)
  - …
- We can use it to *predict* the response of a new instance.
  - How good are our predictions?

Pause: Classification vs. Regression

- Both are “Learn a function from labeled examples.”
- The only difference is the label’s domain.
  Why make the distinction?
  - Historically, they’ve been studied separately
  - The label domain can significantly impact what algorithms will work or not work
- Classification
  - “Separate the data.”
- Regression
  - “Fit the data.”

Unsupervised Learning

- Take clustering for example.
- Input: “features” Output: “label”
  - Features can be symbols, real numbers, etc…
    - [age, height, weight, gender, hair_colour, …]
  - Labels are **not** given *a priori*. (Frequently |L| is given.)
- Each label describes a subset of the data
  - In clustering, examples that are “close” together are grouped
  - So we need to define “close”
  - Labels are represented by “cluster centres”
- In this case, frequently the groups really are the end result.
  They are subjective: Evaluation is difficult.

Reinforcement Learning

- Input: “observations”, “rewards” Output: “actions”
  - Observations may be real or discrete
  - Reward is a real number
  - Actions may be real or discrete
- The situation here is one of an agent (think “robot”) interacting with its environment
- The interaction is continuing -- actions are chosen and performance is measured.
- Performance can be improved (i.e. reward increased.) over time by analyzing past experience.

Okay: Let’s tie these together

- Associations, Classification, Regression, Clustering, Reinforcement Learning
- We’re going to take features, and predict something: label, response, good action
- We’re going to *learn* this predictor from previous data

A Closer Look at Classification

- We will now look at an example classification problem.
- Slides courtesy of Russ Greiner, and Duda, Hart, and Stork.

Intro to Machine Learning
(aka Pattern Recognition)
Chapter 1.1—1.6, Duda, Hart, Stork

Machine Perception
An Example
Pattern Recognition Systems
The Design Cycle
Learning and Adaptation
Conclusion

Machine Perception

Build a machine that can recognize patterns:

- Speech recognition
- Fingerprint identification
- OCR (Optical Character Recognition)
- DNA sequence identification
- …

Example

Sort Fish

into Species

- **Sea bass**
- **Salmon**

using optical sensing

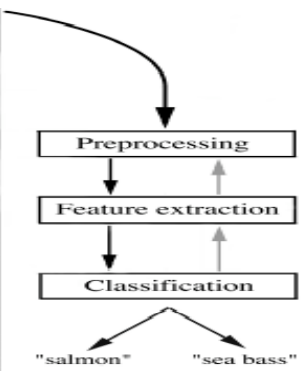
Problem Analysis

- Extract *features* from sample images:
  - Length
  - Width
  - Average pixel brightness
  - Number and shape of fins
  - Position of mouth
  - …

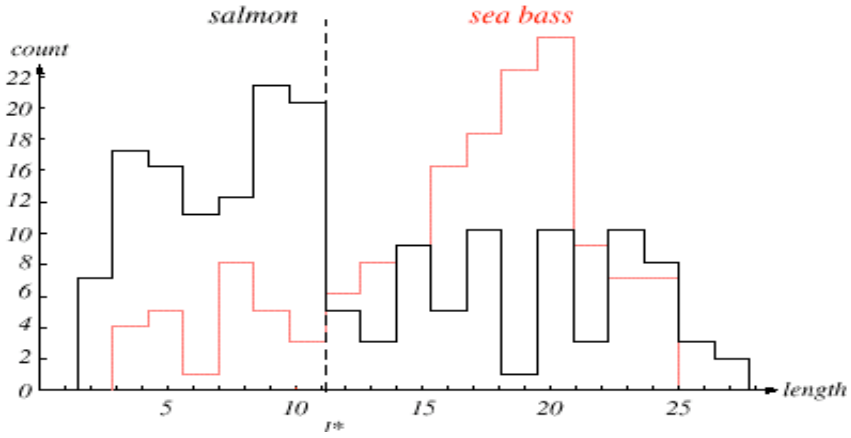
- Classifier makes decision for FishX, based on values of these features!

Preprocessing

- Use *segmentation* to isolate
  - fish from background
  - fish from one another
- Send info about each single fish to *feature extractor*,
  … compresses quantity of data,
  into small set of features
- Classifier sees these features

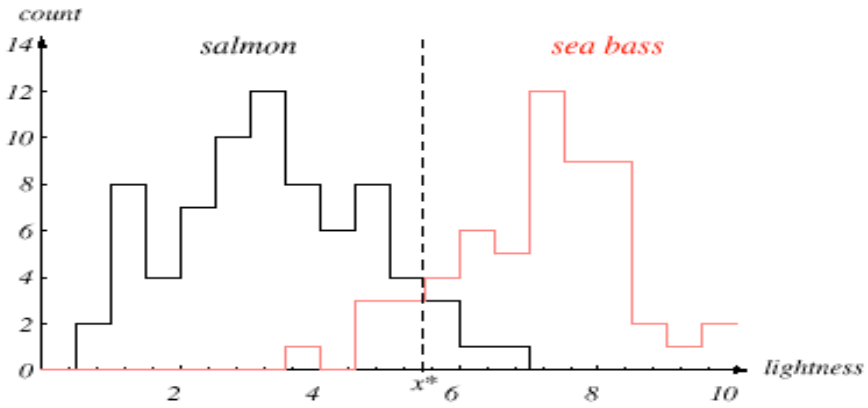


Use "Length"?



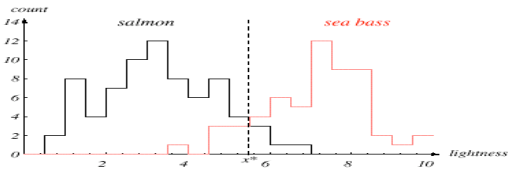
- Problematic… many incorrect classifications

Use “Lightness”?



- Better… fewer incorrect classifications
- Still not perfect

Where to place boundary?



- *Salmon Region* intersects *SeaBass Region*
  - ⇒ So no “boundary” is perfect
    - *Smaller* boundary ⇒ fewer SeaBass classified as Salmon
    - *Larger* boundary ⇒ fewer Salmon classified as SeaBass
- Which is best… depends on misclassification costs

Task of decision theory

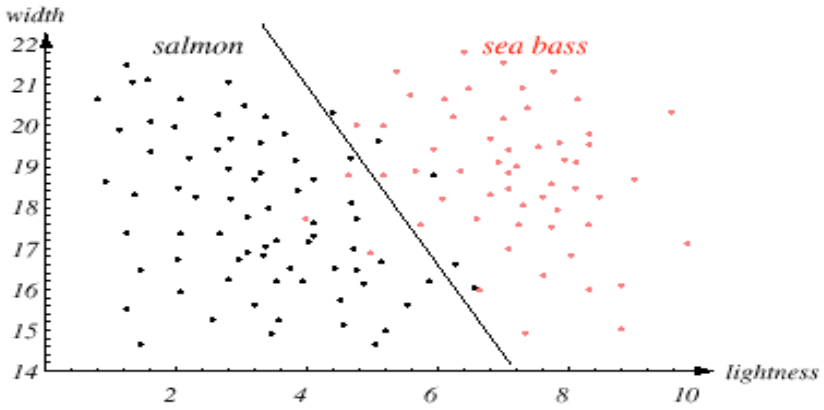
Why not 2 features?

- Use *lightness* and *width* of fish

Fish → $x^T = [x_1, x_2]$

x_1: **Lightness** x_2: **Width**

Results

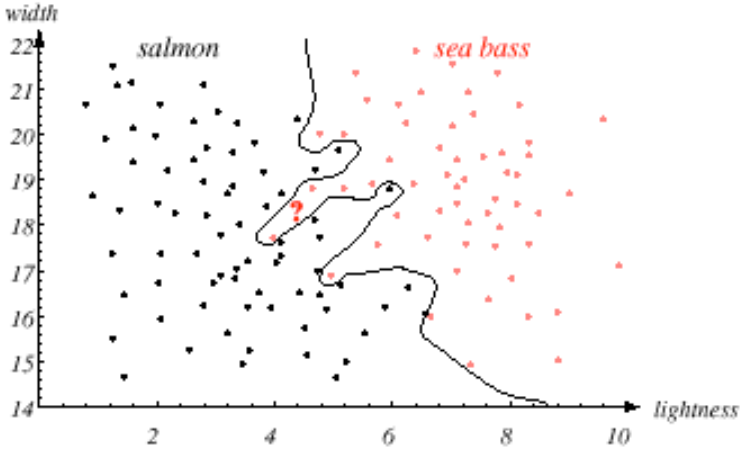


- Much better…
  very few incorrect classifications !

How to produce Better Classifier?

- Perhaps add other features?
  - ideally, not correlated with current features
  - Warning: “noisy features” will ***reduce*** performance
- Best decision boundary ≡ one that provides optimal performance
  - Not necessarily LINE
  - Eg …

“Optimal Performance” ??

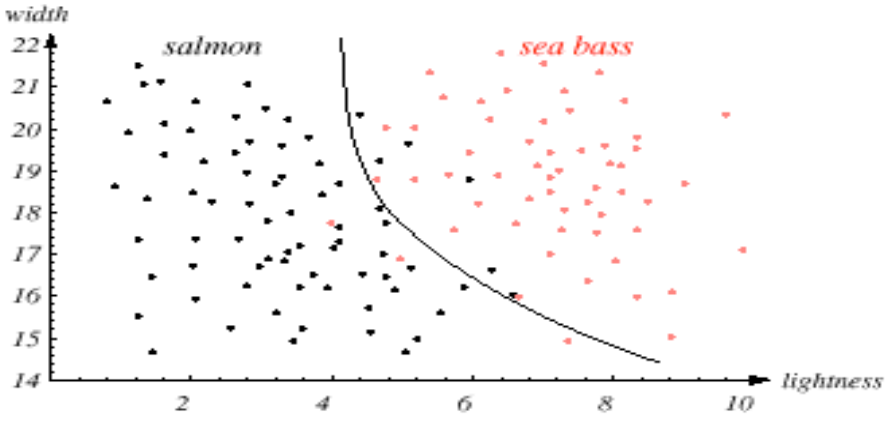


Objective: Handle Novel Data

- Goal:
  - Optimal performance on *NOVEL* data
  - Performance on TRAINING DATA != Performance on NOVEL data

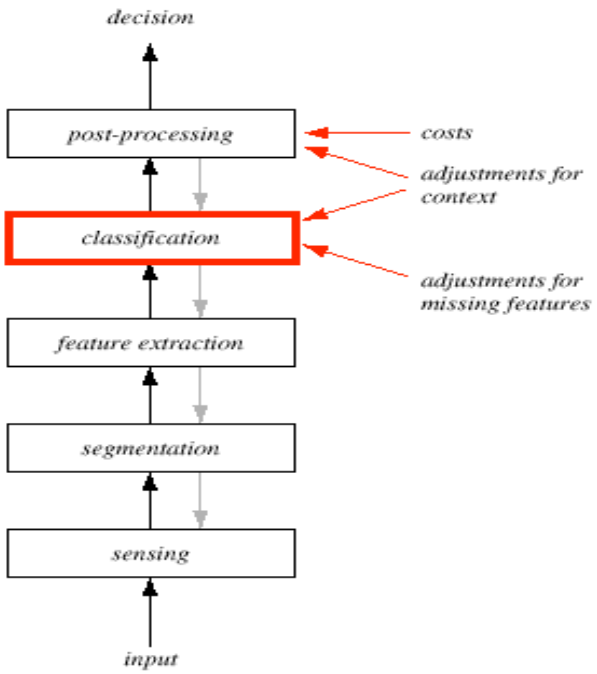
Issue of generalization!

Simple (non-line) Boundary



Pattern Recognition Systems

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



Machine Learning Steps

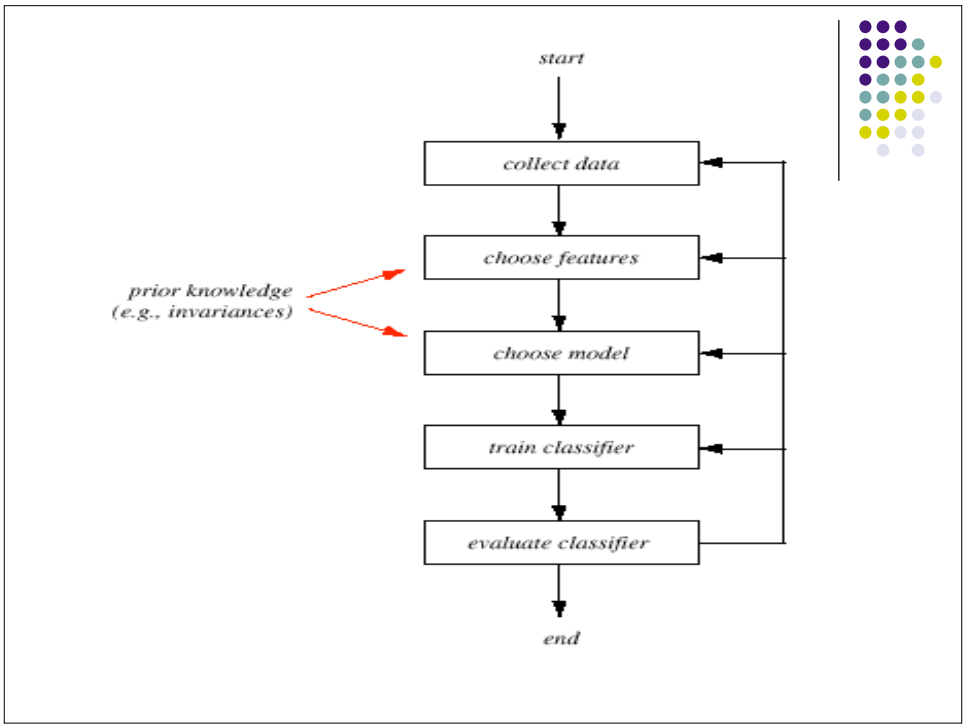
- Feature extraction
  - Discriminative features
  - Want features INVARIANT
    wrt translation, rotation, scale.
- Classification
  - Using feature vector (provided by feature extractor)
    to assign given object to a *category*
- Post Processing
  - Exploit context (information not in the target pattern itself)

  to improve performance

The Design Cycle

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

Computational Complexity



Data Collection

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

Which Features?

- Depends on characteristics of problem domain
- Ideally…
  - Simple to extract
  - Invariant to irrelevant transformation
  - Insensitive to noise

Which Model?

- Try simple one
- If not satisfied with performance consider another class of model

Training

- Use data to obtain good classifier
  - identify best model
  - determine appropriate parameters
- Many procedures for training classifiers and choosing models

Evaluation

- Measure error rate

  ≈ performance
- May suggest switching
  - from one set of features to another one
  - from one model to another

Computational Complexity

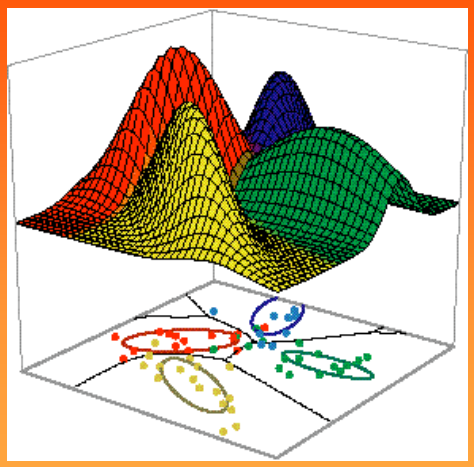
- Trade-off between computational ease and performance?
- How algorithm scales as function of
  - number of features, patterns or categories?

Learning and Adaptation

- Supervised learning
  - A teacher provides a category label or cost for each pattern in the training set
- Unsupervised learning
  - System forms clusters or “natural groupings” of input patterns

Conclusion

- Machine Learning has many challenging sub-problems
- Many of these sub-problems can be solved!
- Many fascinating unsolved problems still remain



Pattern Classification