Machine Learning Overview

Dr. Xiaowei Huang

https://cgi.csc.liv.ac.uk/~xiaowei/

In the last lecture,

- Module Information
- Contents of the module
- What is machine learning?

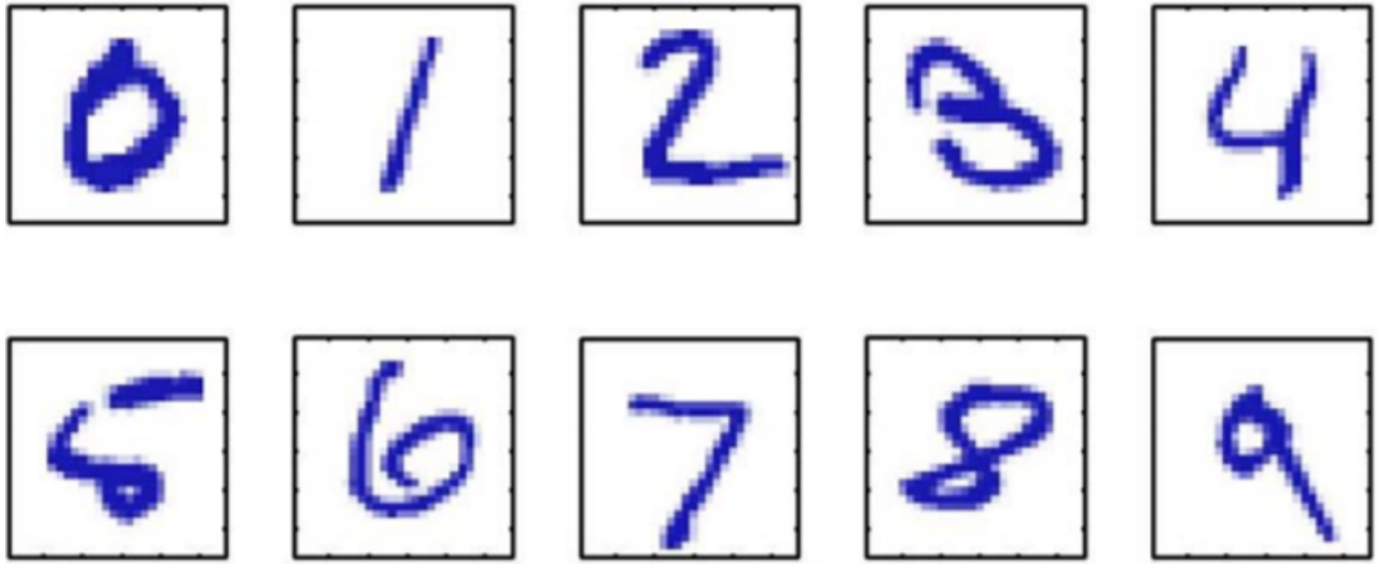
Today’s Content

- A few applications of machine learning
- define the supervised and unsupervised learning tasks
- consider how to represent instances as fixed-length feature vectors
- understand the concepts

Where Machine Learning is used/useful?

- Can be applied in situations where it is very challenging (= impossible) to define rules by hand, e.g.:
  - Face detection
  - Speech recognition
  - Stock prediction

Example 1: hand-written digit recognition



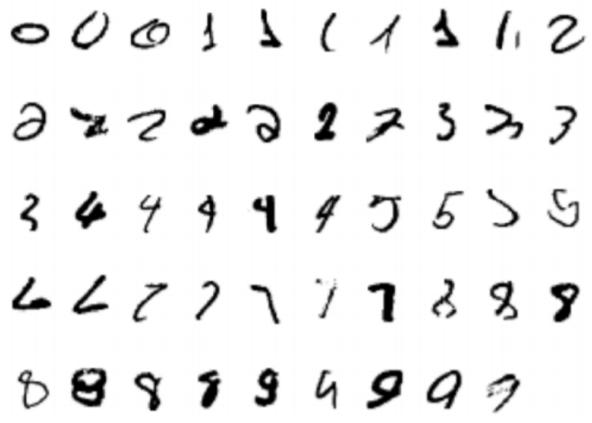
Images are 28 x 28 pixels

Represent input image as a vector $x \in \mathrm{R}^{784}$,
learn a classifier f(x) such that

$$f : \mathrm{R}^{784} \rightarrow \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

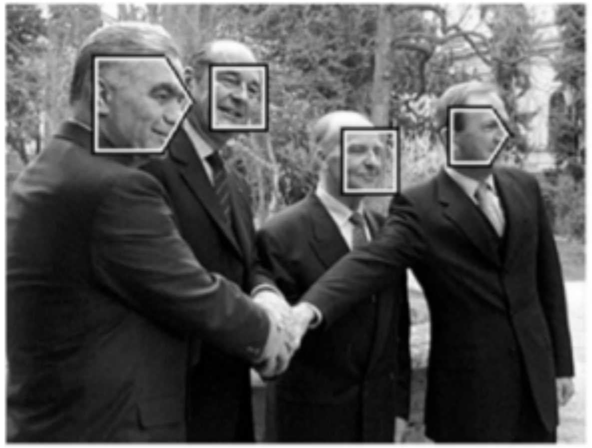
How to proceed …

- As a supervised classification problem
- Start with training data, e.g. 6000 examples of each digit



- Can achieve testing error of 0.4%
- One of the first commercial and widely used ML systems (for zip codes & checks)

Example 2: Face detection

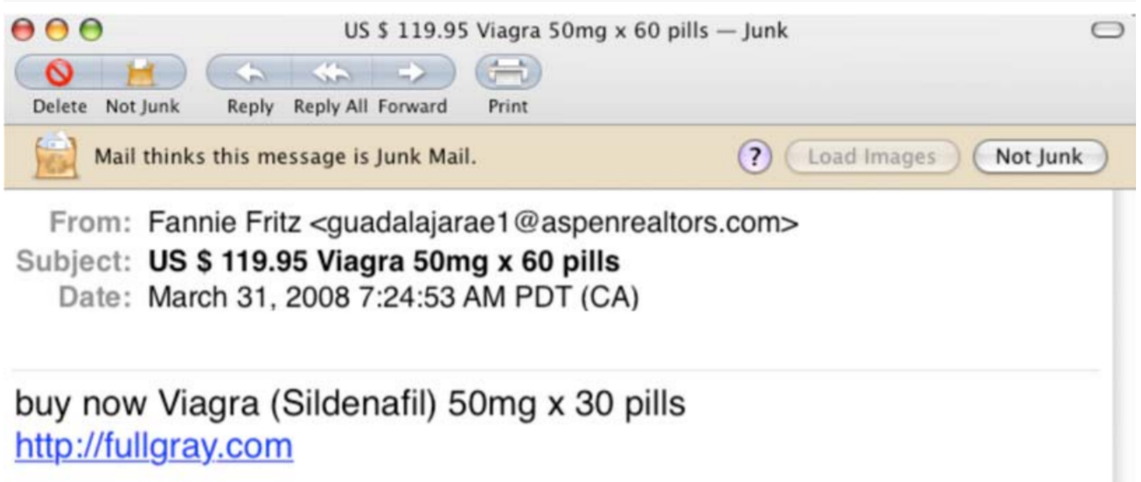


- Again, a supervised classification problem
- Need to classify an image window into three classes:
  - non-face
  - frontal-face
  - profile-face

Classifier is learnt from labelled data

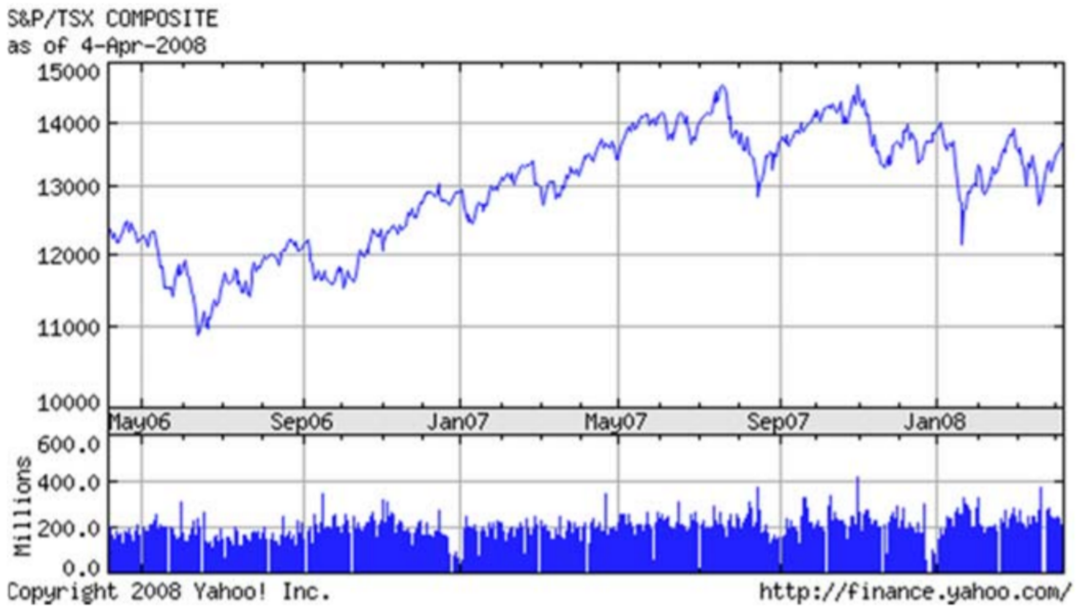
- Training data for frontal faces
  - 5000 faces
    - All near frontal
    - Age, race, gender, lighting
  - 10^8 non faces
  - faces are normalized
    - scale, translation (a **translation** is a geometric **transformation** that moves every point of a figure or a space by the same distance in a given direction)

Example 3: Spam detection



- This is a classification problem
- Task is to classify email into spam/non-spam
- Data x_i is word count, e.g. of viagra, outperform, “you may be surprised to be contacted” …
- Requires a learning system as “enemy” keeps innovating

Example 4: Stock price prediction



- Task is to predict stock price at future date
- This is a regression task, as the output is continuous

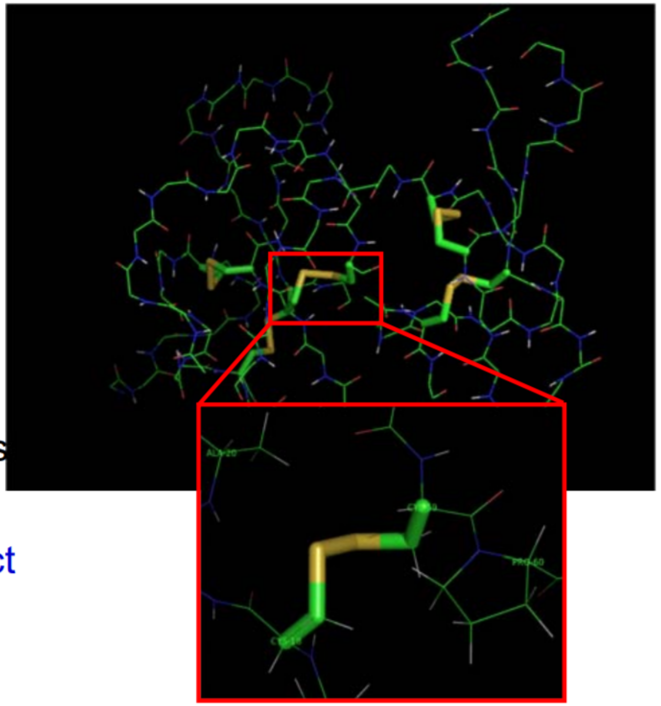
Example 5: Computational biology

x

AVITGACERDLQCG
KGTCCAVSLWIKSV
RVCTPVGTSGEDCH
PASHKIPFSGQRMH
HTCPCAPNLACVQT
SPKKFKCLSK

Protein Structure and Disulfide Bridges

y



Regression task: given sequence predict 3D structure

Protein: 1IMT

- **Protein structure prediction** is the inference of the three-dimensional structure of a protein from its amino acid sequence
- based on the dataset alone, the algorithm can learn how to combine multiple features of the input data into a more abstract set of features from which to conduct further learning

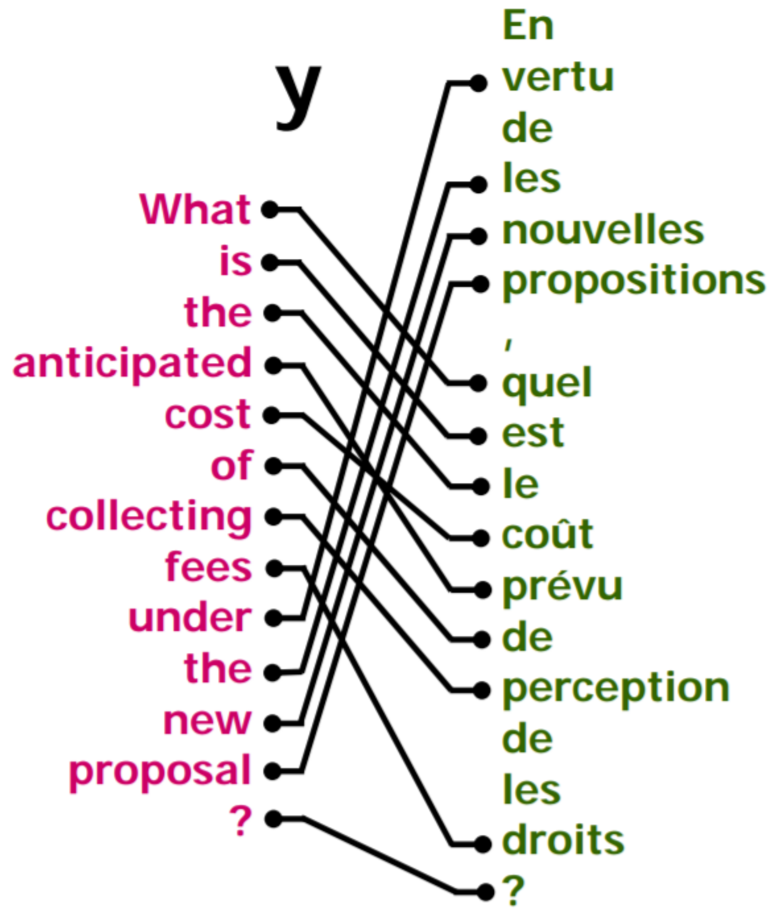
Web examples: Machine translation

Use of aligned text

x

What is the anticipated cost of collecting fees under the new proposal?

En vertu des nouvelles propositions, quel est le coût prévu de perception des droits?



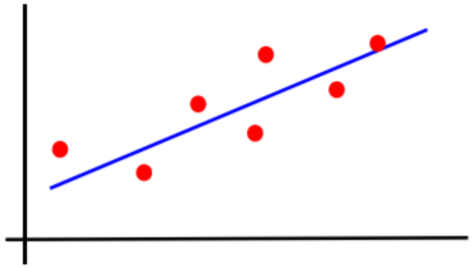
e.g. Google translate

Web examples: Recommender systems

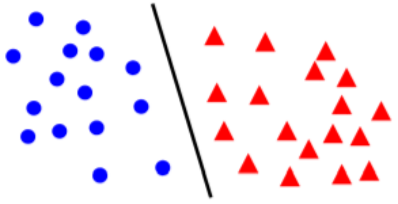
People who bought Hastie …

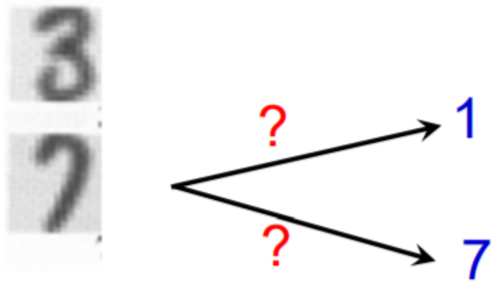
Three canonical learning problems

1. Regression - supervised
   - estimate parameters, e.g. of weight vs height



2. Classification - supervised
   - estimate class, e.g. handwritten digit classification

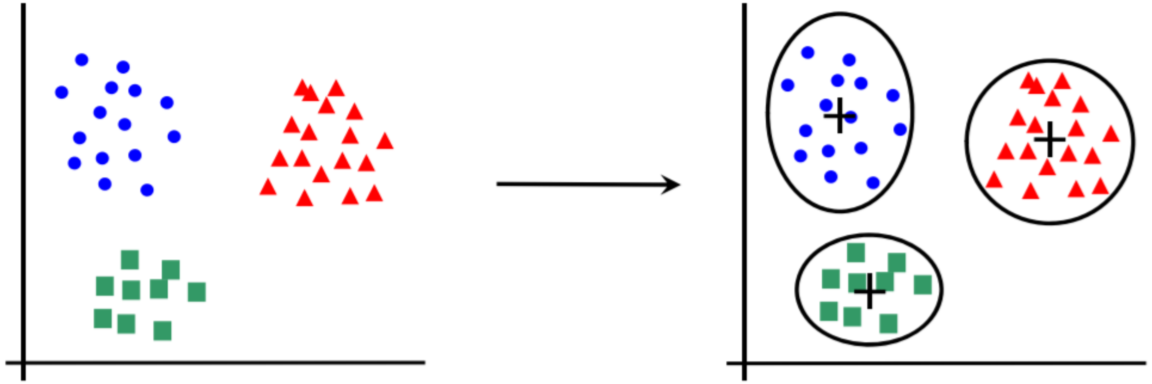




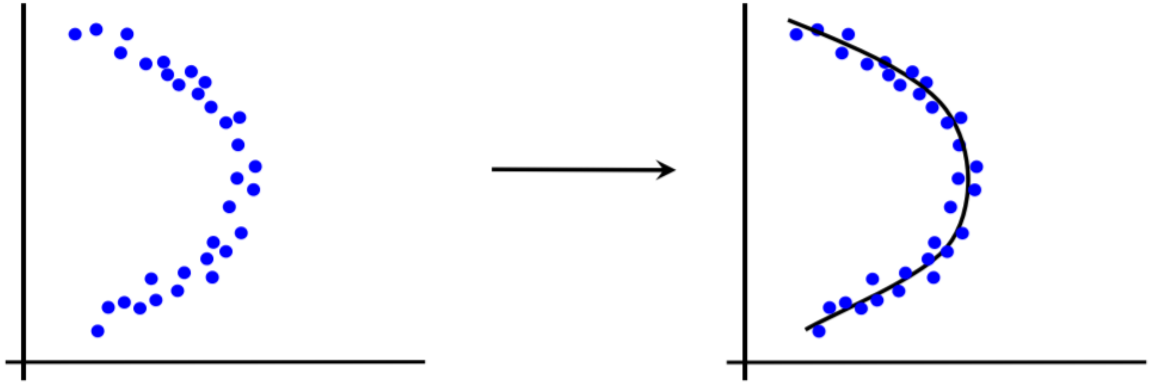
Three canonical learning problems

3. Unsupervised learning – model the data

- clustering



- dimensionality reduction



Can I eat this mushroom?

I don't know what type it is – I've never seen it before. Is it edible or poisonous?

Can I eat this mushroom?

suppose we’re given examples of edible and poisonous mushrooms (we’ll refer to these as *training examples* or *training instances*)

edible




poisonous



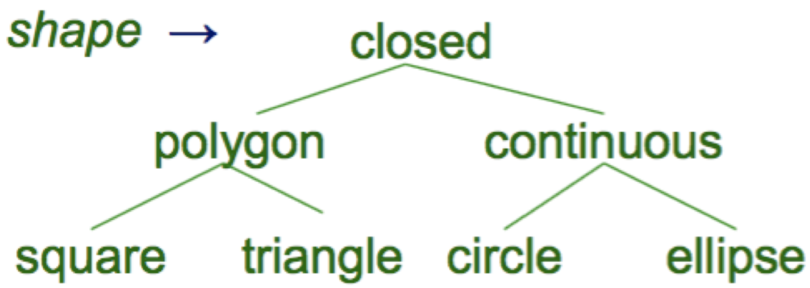

Representing instances using feature vectors

- we need some way to represent each instance
- one common way to do this: use a fixed-length vector to represent *features* (a.k.a. *attributes*) of each instance
- also represent *class label* of each instance

$$
\begin{array}{lllllll}
 & \text{cap-shape} & \text{cap-surface} & \text{cap-color} & \text{bruises} & \text{odor} & \\
\mathbf{x}^{(1)} = \langle & \text{bell}, & \text{fibrous}, & \text{gray}, & \text{false}, & \text{foul}, \ldots \rangle \\
\mathbf{x}^{(2)} = \langle & \text{convex}, & \text{scaly}, & \text{purple}, & \text{false}, & \text{musty}, \ldots \rangle \\
\mathbf{x}^{(3)} = \langle & \text{bell}, & \text{smooth}, & \text{red}, & \text{true}, & \text{musty}, \ldots \rangle \\
 & & \vdots & & & &
\end{array}
\qquad
\begin{array}{l}
\text{class} \\
y^{(1)} = \text{edible} \\
y^{(2)} = \text{poisonous} \\
y^{(3)} = \text{edible} \\
\vdots
\end{array}
$$

Standard feature types

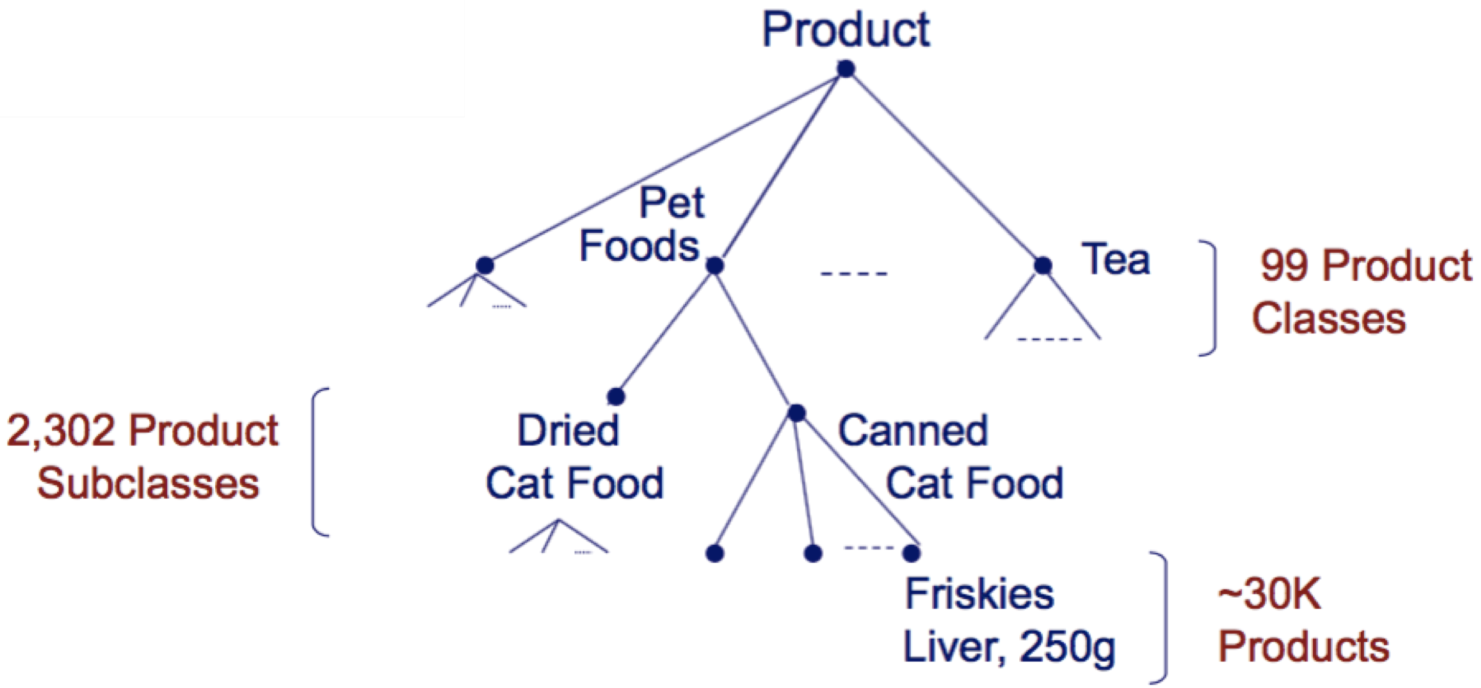
- *nominal* (including Boolean)
  - no ordering among possible values
    - e.g. *color* ∈ {*red, blue, green*} (vs. *color = 1000* Hertz)
- *ordinal*
  - possible values of the feature are totally ordered e.g. *size* ∈ {*small, medium, large*}
- *numeric (continuous)*
  - *E.g., weight* ∈ [0...500]
- *hierarchical*
  – possible values are partially *ordered* in a hierarchy



Feature hierarchy example

- Lawrence et al., *Data Mining and Knowledge Discovery* 5(1-2), 2001

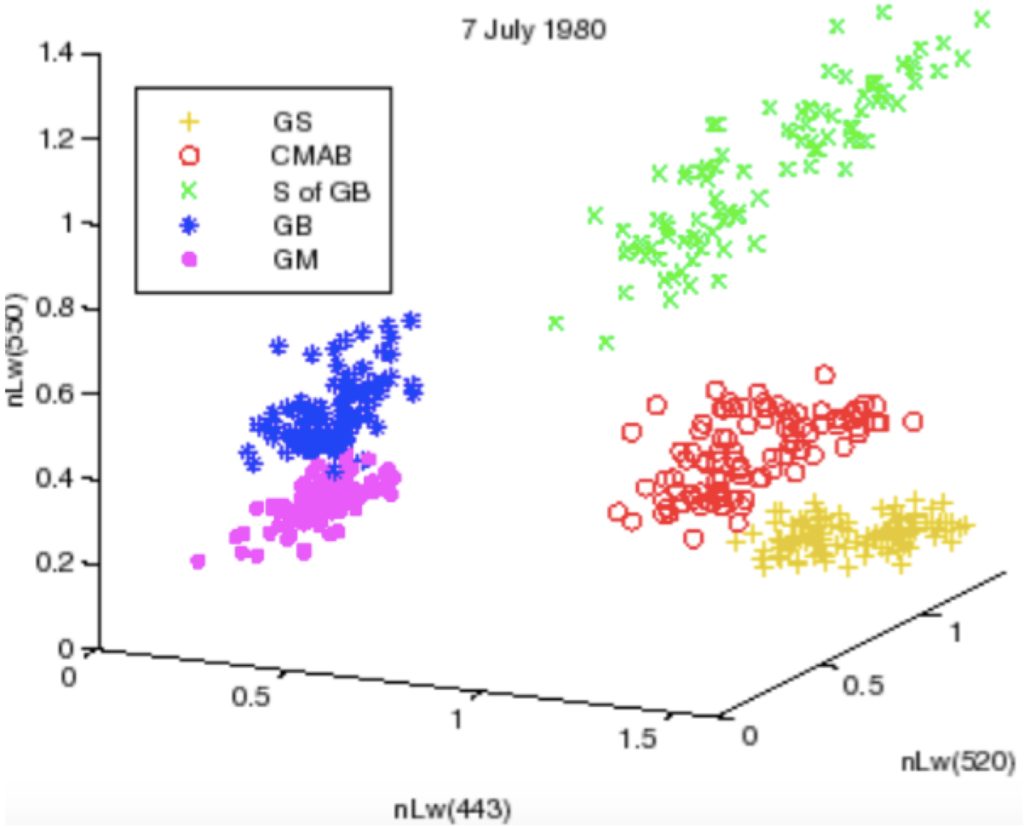
Structure of one feature



Feature space

- we can think of each instance as representing a point in a d-dimensional feature space where d is the number of features

example: optical properties of oceans in three spectral bands [Traykovski and Sosik, *Ocean Optics XIV Conference Proceedings*, 1998]



How about a 3-dimensional space for height, weight, and body fat percentage?

Another view of the feature-vector representation: a single database table

	feature 1	feature 2	. . .	feature d	class
instance 1	0.0	small		red	true
instance 2	9.3	medium		red	false
instance 3	8.2	small		blue	false
. . .					
instance n	5.7	medium		green	true

The supervised learning task

- problem setting
  - set of possible instances: *X*
  - unknown *target function:* $f : X \rightarrow Y$
  - set of *models* (a.k.a. *hypotheses*): $H = \{h \mid h : X \rightarrow Y\}$

- given *training set* of instances of unknown target function f

$$\left(\mathbf{x}^{(1)}, y^{(1)}\right), \left(\mathbf{x}^{(2)}, y^{(2)}\right) \ldots \left(\mathbf{x}^{(m)}, y^{(m)}\right)$$

- Output
  - model $h \in H$ that best approximates target function

The supervised learning task

- when y is discrete, we term this a *classification* task (or *concept learning*)

- when y is continuous, it is a *regression* task

- there are also tasks in which each y is more structured object like a *sequence* of discrete labels (as in e.g. image segmentation, machine translation)

Batch vs. online learning

- In batch learning, the learner is given the training set as a batch (i.e. all at once)

$$\left(\mathbf{x}^{(1)}, y^{(1)}\right), \left(\mathbf{x}^{(2)}, y^{(2)}\right) \dots \left(\mathbf{x}^{(m)}, y^{(m)}\right)$$

- In online learning, the learner receives instances sequentially, and updates the model after each (for some tasks it might have to classify/make a prediction for each x(i) before seeing y(i))

$$\left(\mathbf{x}^{(1)}, y^{(1)}\right) \qquad \left(\mathbf{x}^{(2)}, y^{(2)}\right) \qquad \left(\mathbf{x}^{(i)}, y^{(i)}\right)$$

time