Introduction to Machine Learning

CS195-5-2003

Thomas Hofmann

What is Machine Learning ?

- Machine learning deals with the design of computer programs and systems that are able to take advantage of data, examples or experiences to improve their accuracy or performance on a specific task or set of tasks.
- Tasks: decision problems, control problems or prediction problems

Abstraction Levels

Class of problems a learning machine can handle.

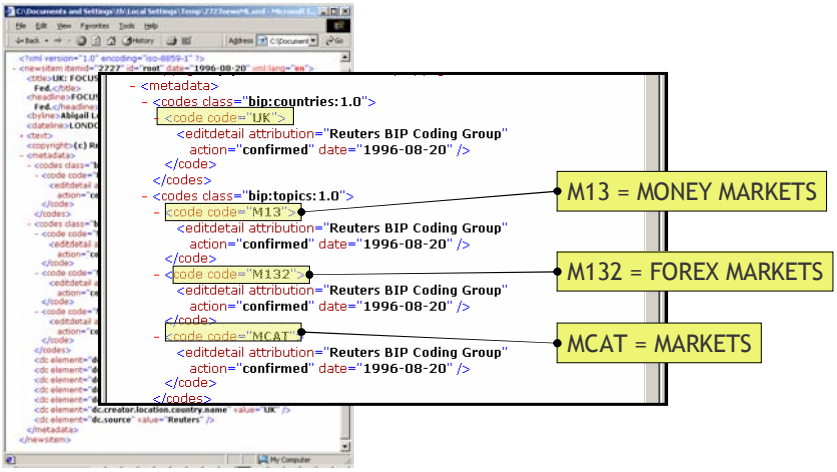
Training data
(indirect specification)

Specific problem from this class of problems.

Test data

Specific instance for which a solution needs to be computed

Automatic Document Categorization & Annotation



Categorization With Expert Rules

Expert

```
if contains('yen')  or
   contains('euro')
then label=M132
```

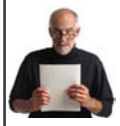
M132 = FOREX MARKETS

Problem: Low coverage, moderate accuracy, difficulty of formalizing expert knowledge

Example-Based Categorization

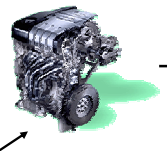
Training Examples

M132 = FOREX MARKETS



Expert

Training



Learning Machine

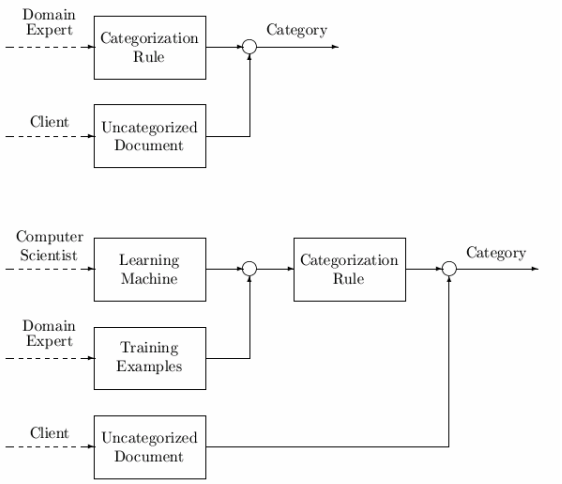
Inductive Inference

/* some 'complicated' algorithm */

Recall

M132 = FOREX MARKETS

Direct & Indirect Approach to Categorization



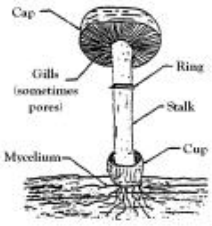
Living Longer With Machine Learning (1)

The Challenge

- Imagine that you visit an unknown country far away from here, where people use a large variety of mushrooms in their everyday diet.
- Some of the mushrooms are poisonous while some are edible. There is obviously some incentive to knowing which one is which.

- Luckily, you have a digital mushroom field guide available that is uploaded to your handheld computer.
- Huge table of mushroom descriptions along with the crucial bit of information…

Living Longer With Machine Learning (2)



The Data

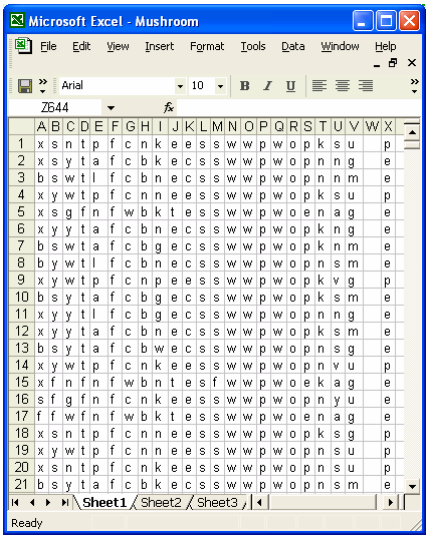
- Here are some of the attributes:

1	cap-shape	bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
2	cap-surface	fibrous=f, grooves=g, scaly=y, smooth=s
3	cap-color	brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
4	bruises	bruises=t, no=f
5	odor	almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
6	gill-attachment	attached=a, descending=d, free=f, notched=n
...		
22	habitat	grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

Living Longer With Machine Learning (3)

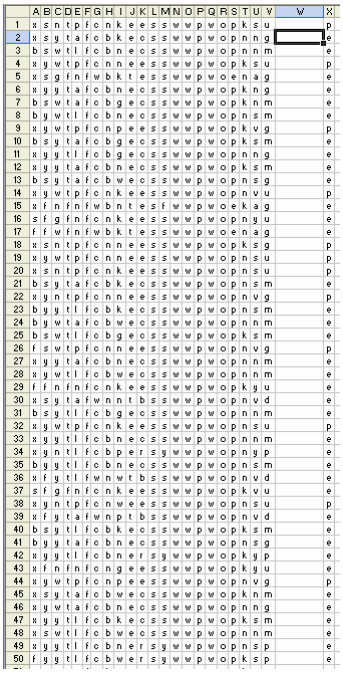
The Data

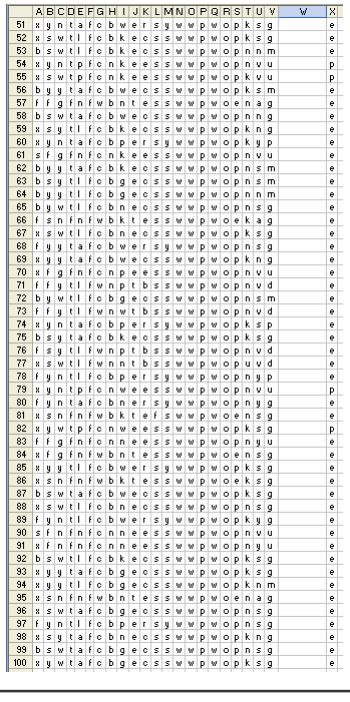
- Here is what the table looks like



- Time for a little entrance examination …

The training data





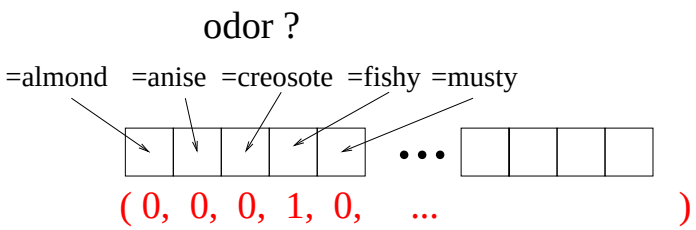
The test data

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
f	y	g	t	n	f	c	b	n	t	b	s	s	g	g	p	w	o	p	k	y	d	
x	y	g	f	f	f	c	b	p	e	b	k	k	b	p	p	w	o	l	h	v	d	
f	y	n	t	n	f	c	b	p	t	b	s	s	p	w	p	w	o	p	k	v	d	
f	y	n	t	n	f	c	b	p	t	b	s	s	g	g	p	w	o	p	k	v	d	
x	s	g	f	c	f	w	n	n	e	b	s	s	w	w	p	w	o	p	k	s	d	
f	y	g	t	n	f	c	b	p	t	b	s	s	p	w	p	w	o	p	n	y	d	
f	y	e	t	n	f	c	b	w	t	b	s	s	w	w	p	w	o	p	n	v	d	
x	f	g	f	f	f	c	b	g	e	b	k	k	p	n	p	w	o	l	h	y	d	
x	y	n	t	n	f	c	b	w	t	b	s	s	g	p	p	w	o	p	n	y	d	
f	y	e	t	n	f	c	b	u	t	b	s	s	p	g	p	w	o	p	n	v	d	
f	f	e	t	n	f	c	b	n	t	b	s	s	p	g	p	w	o	p	k	v	d	
x	s	p	f	c	f	c	n	n	e	b	s	s	w	w	p	w	o	p	n	s	d	
x	f	g	f	f	f	c	b	h	e	b	k	k	p	p	p	w	o	l	h	v	d	
f	y	e	t	n	f	c	b	p	t	b	s	s	p	w	p	w	o	p	k	v	d	
f	y	g	t	n	f	c	b	p	t	b	s	s	g	w	p	w	o	p	k	v	d	
x	s	p	f	c	f	c	n	p	e	b	s	s	w	w	p	w	o	p	n	v	d	
f	f	g	t	n	f	c	b	u	t	b	s	s	w	g	p	w	o	p	n	y	d	
f	f	e	t	n	f	c	b	n	t	b	s	s	w	w	p	w	o	p	k	y	d	
f	f	n	t	n	f	c	b	u	t	b	s	s	g	w	p	w	o	p	k	v	d	
x	y	g	t	n	f	c	b	n	t	b	s	s	w	w	p	w	o	p	n	v	d	
x	y	n	t	n	f	c	b	w	t	b	s	s	w	p	p	w	o	p	n	y	d	
f	y	n	t	n	f	c	b	u	t	b	s	s	g	p	p	w	o	p	n	y	d	
x	f	p	f	c	f	c	n	p	e	b	s	s	w	w	p	w	o	p	k	v	d	

Living Longer With Machine Learning (4)

The Representation

- Multi-valued attributes are mapped to binary representation (also known as *orthogonal encoding*)



Living Longer With Machine Learning (5)

The Representation

x s n t p f c n k e e s s w w p w o p k s u

0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0

Discriminant Function

- Formally (for binary inputs)

$$f : \{0,1\}^d \rightarrow \{-1,1\}$$

Edible or not

- More generally:

$$f : \mathcal{X} \subseteq \Re^d \rightarrow \{-1,1\}$$

- Maps a pattern to its class or label

$\mathbf{x}$ $\mathbf{y}$

Linear Discriminant

- Linear discriminant

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{d} x_i w_i + b\right), \quad w_i, b \in \Re\,.$$

- Intuitively, each attribute which is present contributes with a weight that may be positive or negative.
- A positive weight indicates that the attribute provides evidence for the mushroom being edible, a negative weight provides evidence for the opposite hypothesis.
- The magnitude of the weight determines the influence of this attribute relative to other attributes.
- The bias specifies the (negative of the) threshold for making the final decision based on the accumulative evidence.

Learning Linear Discriminant

- Goal: Adjust weights and bias in way that maximizes the classification accuracy.
- Problem has been restricted to a pre-specified hypothesis class (linear discriminants)
- Learning procedure = parameter fitting

Perceptron Algorithm (1)

- Initialize the weights & bias
- Cycle through the training data
- Test whether current example is correctly classified
- If not, perform an update step by adjusting w,b. **Learning from mistakes.**
- Until all training data are correctly classified

Perceptron Algorithm (2)

Algorithm 1 Perceptron algorithm

```
1: w ← 0, b ← 0
2: repeat
3:    errors ← 0
4:    /* cycle through all training examples */
5:    for i = 1, ..., n do
6:       compute f(x_i) = sign(⟨w, x_i⟩ + b)
7:       if f(x_i) ≠ y_i then
8:          w ← w + y_i x_i
9:          b ← b + y_i
10:         errors ← errors + 1
11:      end if
12:   end for
13: until error = 0
14: output w, b
```


Matlab (1)

Matlab 1 Reading in a matlab data file.

```
% load data
load 'mushroom';

% get dimensions
[dim num] = size(features);
pos       = length(find(labels==1));
neg       = num - pos ;

% print some info
fprintf('Number of training data  = %d\n',num);
fprintf('Pos/neg examples         = %d / %d \n', pos, neg);
fprintf('Number of input features = %d\n',dim);
```

Matlab (2)

Creating zero vectors, no variable declaration

for loop

Matrix indexing, transposing, multiplications

fprintf

Matlab 2 Perceptron algorithm, primal form.

```
% Initialize parameters
w = zeros(dim,1);
b = zeros(1,1);

% Perceptron learning, main loop
max_iter = 100;
for iter=1:max_iter
   errors = 0;
   for i=1:num
      if ( labels(i) * ( w' * features(:,i) + b ) <= 0 )
         w = w + labels(i) * features(:,i);
         b = b + labels(i);
         errors = errors + 1;
      end
   end
   fprintf('Iteration %d, errors = %d\n', iter, errors);
   if (errors=0)
      break;
   end
end
```


Generalization

- Assuming the perceptron converges, what do we know about the solution?
- It classifies all training examples correctly!
- Yet, how will the discriminant function perform on unseen new data? (Generalization)
- Inductive inference

How can we derive general rules that apply to an infinite number of cases from a finite set of examples?

Empirical vs. Expected Risk

- Empirical risk (based on zero-one loss)

$$J_{emp}(f) = \frac{1}{2n} \sum_{i=1}^{n} |f(\mathbf{x}_i) - y_i|$$

- Expected risk

$$J(f) = \frac{1}{2} \int |f(\mathbf{x}) - y| \, dP(\mathbf{x}, y)$$

Underlying probability law that generates the data.

Estimating Generalization Performance

- Statistical learning theory: derive bounds of the type

$$J(f^*) \leq J_{emp}(f^*) + \text{some function of } n \text{ and } \mathcal{H}$$

- Empirical estimate via hold out data
  - Leave out some random subset of the examples during training (e.g. 20%)
  - Test of these hold-out examples after training

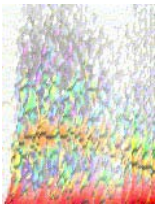
Applications of Pattern Recognition

- Optical character recognition
  - Recognize hand written characters
  - *Zip code recognition in mail sorting facailities*
  - *Reading numbers on checks*

- Object recognition
  - *Recognize object types in images*
  - *Face detection, recognizing animals, cars, etc.*
  - *Image annotation for content-based retrieval*

Applications of Pattern Recognition

- Automatic speech recognition
  - Transcribing speech signals into written text
  - Usually: mapping windows (10ms) of speech to phonemes (approx. 20 different sound classes in English)
  - *Use in voice computer interfaces, calling centers, etc.*



- Text Categorization
  - Annotating documents with content descriptors (topics, categories, taxonomies)
  - *Content management, filtering, routing*

Applications of Pattern Recognition

- Medical diagnosis
  - Prediction based on clinical measurement (lab tests)
  - *Disease prediction or decision about treatment*

Classification & Regression

- Concept learning

$$f : \Re^d \rightarrow \{-1, 1\}$$

dichotomy

- Multi-class classification

$$f : \Re^d \rightarrow \{1, \ldots, K\}$$

finite number of responses = qualitative

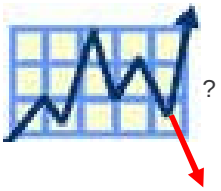
- Regression

$$f : \Re^d \rightarrow \Re$$

quantitative

Applications of Regression

- Financial forecasting
  - Given the past values, observe the next value in a series
  - *Stock market prediction*
  - *Time series prediction*

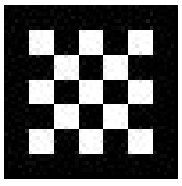


- Optimal Control
  - Given direct or indirect observations of the state of a system and a desired state or trajectory: compute optimal control output
  - Robot arms, Space crafts, etc.

Applications of Regression

- Drug Design
  - *Predict chemical and biological properties of new compounds*

- Game Heuristics
  - *Evaluate game positions (approximately) for use in a heuristic search procedure*
  - *Map positions to a score*

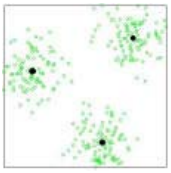


Unsupervised Learning

- Supervised learning (classification, regression)
  - Input patterns along with target values are given as training data
  - Targets can be class labels or numerical values
  - Find (stochastic or deterministic) mapping from inputs to outputs
- Unsupervised learning
  - Only a set of input patterns is given

Unsupervised Learning

- Structure detection
  - Data clustering
    - finding groups of similar patterns
    - Document clustering, protein clustering, texture-based image segmentation
  - Dimension reduction
    - Finding low-dimensional subspaces or data manifolds
  - Data mining
    - Discovering unexpected regularities



Unsupervised Learning

- Probabilistic modeling (density estimation)
  - Medical diagnosis:
    - learn about dependencies between symptoms and diseases
    - *E.g. Bayesian networks*

  - Data compression
    - Shannon: optimal expected codeword length E[-log p(x)] = entropy
    - For example: image compression

The End