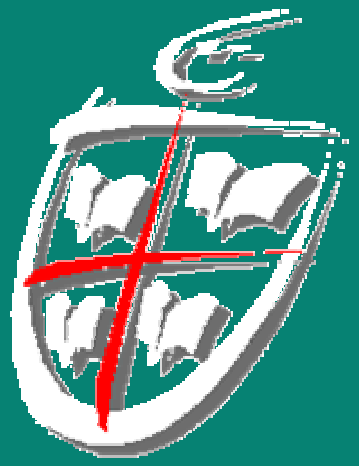


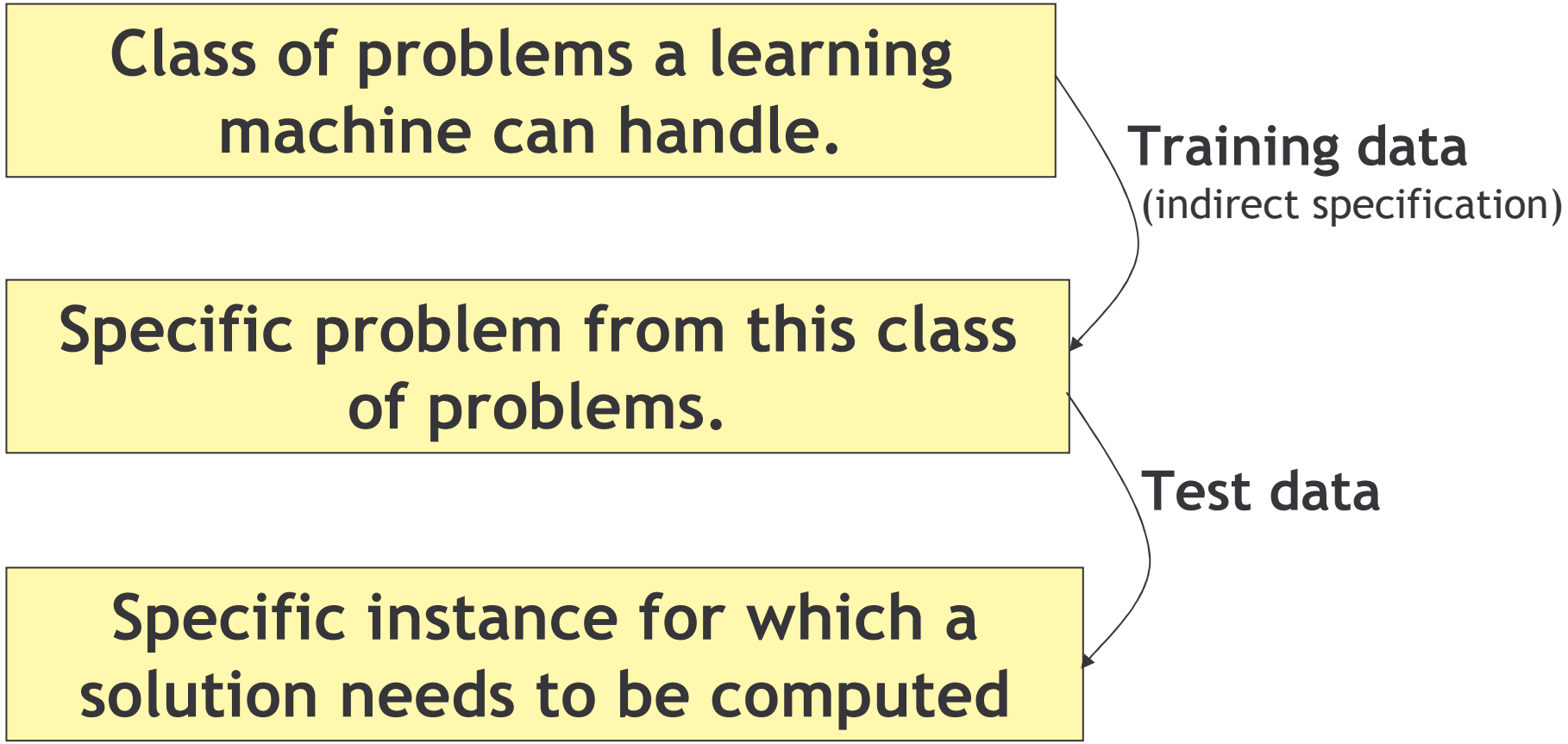
CS900-3a
Computer Science –
A Multifaceted Introduction

Lecture #10
A Computer Learns to Read

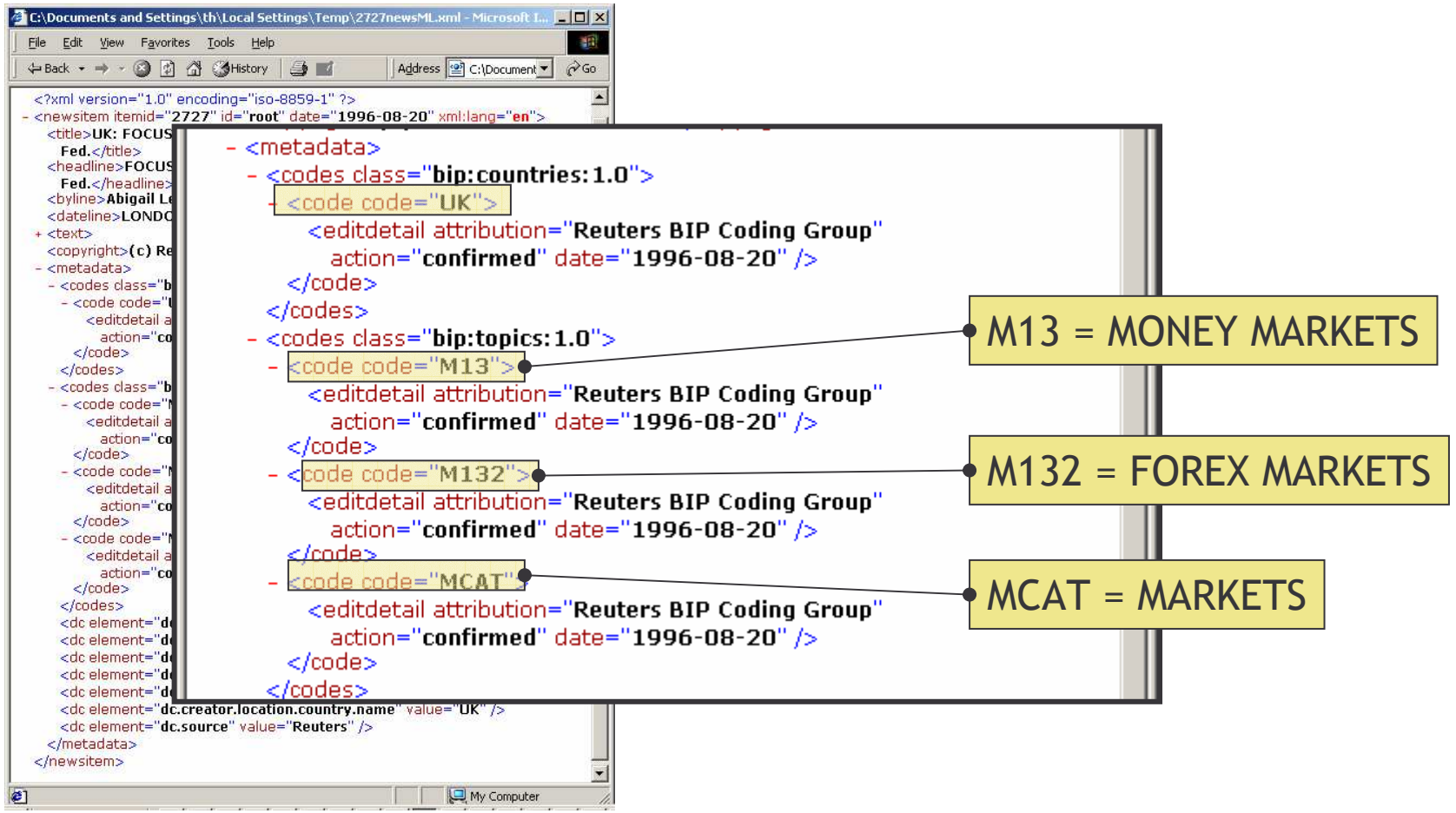
What is Machine Learning?

- 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: categorization problems, decision problems, control problems, prediction problems, etc.

Abstraction Levels



Application: Text Categorization



Categorization With Expert Rules

Expert

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

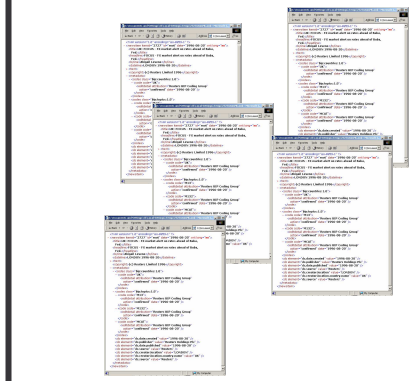
M132 = FOREX MARKETS

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

Example-Based Categorization

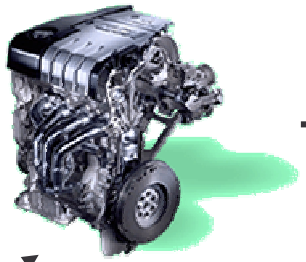
Training Examples

M132 = FOREX MARKETS



Expert

Training



Learning Machine

Inductive Inference

/* some 'complicated' algorithm */

Recall

M132 = FOREX MARKETS

Living Healthy With Machine Learning

The Challenge

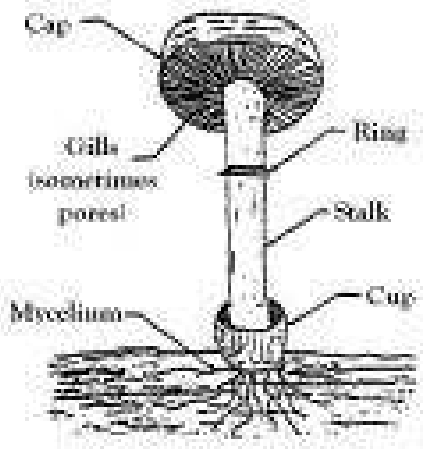
- Imagine that you visit an unknown country far away from here, where people use a large variety of mushrooms in their everyday diet.
- Some of the mushrooms are poisonous while some are edible. There is obviously some incentive to knowing which one is which.

- 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 Healthy With Machine Learning

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 Healthy With Machine Learning

The Data

- Here is what the table looks like

- Time for a little entrance examination …

Microsoft Excel - Mushroom

File Edit View Insert Format Tools Data Window Help

Arial 10 B I U

Z644

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	s	u		p
2	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	n	g		e
3	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m		e
4	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	u		p
5	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g		e
6	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	n	g		e
7	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	n	m		e
8	b	y	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	m		e
9	x	y	w	t	p	f	c	n	p	e	e	s	s	w	w	p	w	o	p	k	v	g		p
10	b	s	y	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	m		e
11	x	y	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	n	g		e
12	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	s	m		e
13	b	s	y	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	s	g		e
14	x	y	w	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	v	u		p
15	x	f	n	f	n	f	w	b	n	t	e	s	f	w	w	p	w	o	e	k	a	g		e
16	s	f	g	f	n	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	y	u		e
17	f	f	w	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g		e
18	x	s	n	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	g		p
19	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	s	u		p
20	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	s	u		p
21	b	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	s	m		e

Sheet1 Sheet2 Sheet3

Ready

The training data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	s	u		p
2	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	n	g		e
3	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m		e
4	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	u		p
5	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g		e
6	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	n	g		e
7	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	n	m		e
8	b	y	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	m		e
9	x	y	w	t	p	f	c	n	p	e	e	s	s	w	w	p	w	o	p	k	v	g		p
10	b	s	y	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	m		e
11	x	y	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	n	g		e
12	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	s	m		e
13	b	s	y	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	s	g		e
14	x	y	w	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	v	u		p
15	x	f	n	f	n	f	w	b	n	t	e	s	f	w	w	p	w	o	e	k	a	g		e
16	s	f	g	f	n	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	y	u		e
17	f	f	w	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g		e
18	x	s	n	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	g		p
19	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	s	u		p
20	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	s	u		p
21	b	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	s	m		e
22	x	y	n	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	v	g		p
23	b	y	y	t	l	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	s	m		e
24	b	y	w	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	n	m		e
25	b	s	w	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	m		e
26	f	s	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	v	g		p
27	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m		e
28	x	y	w	t	l	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	n	m		e
29	f	f	n	f	n	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	y	u		e
30	x	s	y	t	a	f	w	n	n	t	b	s	s	w	w	p	w	o	p	n	v	d		e
31	b	s	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	n	m		e
32	x	y	w	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	s	u		p
33	x	y	y	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m		e
34	x	y	n	t	l	f	c	b	p	e	r	s	y	w	w	p	w	o	p	n	y	p		e
35	b	y	y	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	m		e
36	x	f	y	t	l	f	w	n	w	t	b	s	s	w	w	p	w	o	p	n	v	d		e
37	s	f	g	f	n	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	v	u		e
38	x	y	n	t	p	f	c	n	w	e	e	s	s	w	w	p	w	o	p	n	s	u		p
39	x	f	y	t	a	f	w	n	p	t	b	s	s	w	w	p	w	o	p	n	v	d		e
40	b	s	y	t	l	f	c	b	k	e	c	s	s	w	w	p	w	o	p	k	s	m		e
41	b	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	g		e
42	x	y	y	t	l	f	c	b	n	e	r	s	y	w	w	p	w	o	p	k	y	p		e
43	x	f	n	f	n	f	c	n	g	e	e	s	s	w	w	p	w	o	p	k	y	u		e
44	x	y	w	t	p	f	c	n	p	e	e	s	s	w	w	p	w	o	p	n	v	g		p
45	x	s	y	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	k	n	m		e
46	x	y	w	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	g		e
47	x	y	y	t	l	f	c	b	k	e	c	s	s	w	w	p	w	o	p	k	s	m		e
48	x	s	w	t	l	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	n	m		e
49	x	y	y	t	l	f	c	b	n	e	r	s	y	w	w	p	w	o	p	n	s	p		e
50	f	y	y	t	l	f	c	b	w	e	r	s	y	w	w	p	w	o	p	k	s	p		e

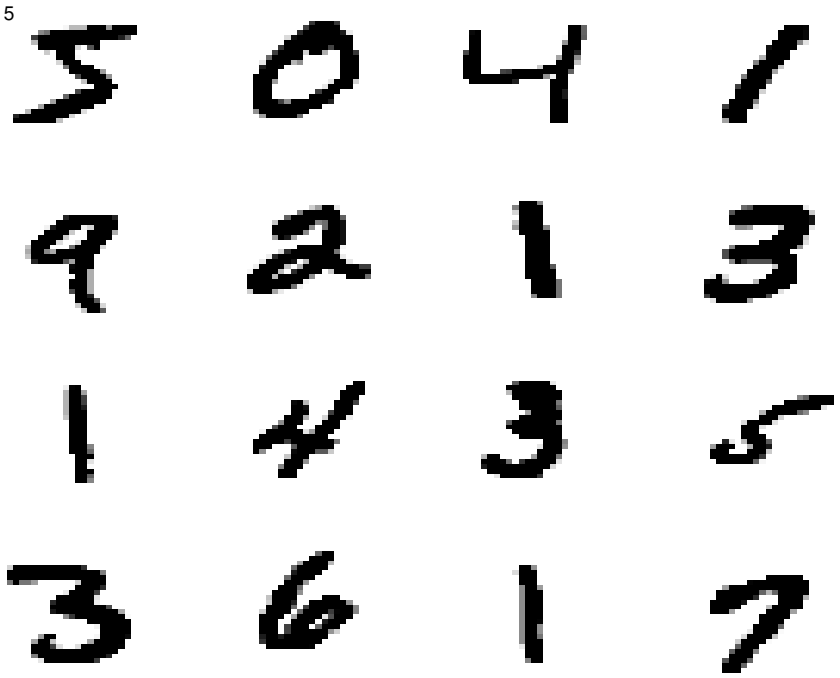
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
51	x	y	n	t	a	f	c	b	w	e	r	s	y	w	w	p	w	o	p	k	s	g		e
52	x	s	w	t	l	f	c	b	k	e	c	s	s	w	w	p	w	o	p	k	s	g		e
53	b	s	w	t	l	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	n	m		e
54	x	y	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	v	u		p
55	x	s	w	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	v	u		p
56	b	y	y	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	k	s	m		e
57	f	f	g	f	n	f	w	b	n	t	e	s	s	w	w	p	w	o	e	n	a	g		e
58	b	s	w	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	n	g		e
59	x	s	y	t	l	f	c	b	k	e	c	s	s	w	w	p	w	o	p	k	n	g		e
60	x	y	n	t	a	f	c	b	p	e	r	s	y	w	w	p	w	o	p	k	y	p		e
61	s	f	g	f	n	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	v	u		e
62	b	y	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	s	m		e
63	b	s	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	s	m		e
64	b	y	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	n	m		e
65	b	y	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	g		e
66	f	s	n	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	k	a	g		e
67	x	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	s	g		e
68	f	y	y	t	a	f	c	b	w	e	r	s	y	w	w	p	w	o	p	n	s	g		e
69	x	y	y	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	k	n	g		e
70	x	f	g	f	n	f	c	n	p	e	e	s	s	w	w	p	w	o	p	n	v	u		e
71	f	f	y	t	l	f	w	n	p	t	b	s	s	w	w	p	w	o	p	n	v	d		e
72	b	y	w	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	s	m		e
73	f	f	y	t	l	f	w	n	w	t	b	s	s	w	w	p	w	o	p	n	v	d		e
74	x	y	n	t	a	f	c	b	p	e	r	s	y	w	w	p	w	o	p	k	s	p		e
75	b	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	k	s	g		e
76	f	s	y	t	l	f	w	n	p	t	b	s	s	w	w	p	w	o	p	n	v	d		e
77	x	s	w	t	l	f	w	n	n	t	b	s	s	w	w	p	w	o	p	u	v	d		e
78	f	y	n	t	l	f	c	b	p	e	r	s	y	w	w	p	w	o	p	n	y	p		e
79	x	y	n	t	p	f	c	n	w	e	e	s	s	w	w	p	w	o	p	n	v	u		p
80	f	y	n	t	a	f	c	b	n	e	r	s	y	w	w	p	w	o	p	n	y	g		e
81	x	s	n	f	n	f	w	b	k	t	e	f	s	w	w	p	w	o	e	n	s	g		e
82	x	y	w	t	p	f	c	n	w	e	e	s	s	w	w	p	w	o	p	k	s	g		p
83	f	f	g	f	n	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	y	u		e
84	x	f	g	f	n	f	w	b	n	t	e	s	s	w	w	p	w	o	e	n	s	g		e
85	x	y	y	t	l	f	c	b	w	e	r	s	y	w	w	p	w	o	p	k	s	g		e
86	x	s	n	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	k	s	g		e
87	b	s	w	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	k	s	g		e
88	x	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	g		e
89	f	y	n	t	l	f	c	b	w	e	r	s	y	w	w	p	w	o	p	k	y	g		e
90	s	f	n	f	n	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	v	u		e
91	x	f	n	f	n	f	c	n	n	e	e	s	s	w	w	p	w	o	p	n	y	u		e
92	b	s	w	t	l	f	c	b	k	e	c	s	s	w	w	p	w	o	p	k	s	g		e
93	x	y	y	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	g		e
94	x	y	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	n	m		e
95	x	s	n	f	n	f	w	b	n	t	e	s	s	w	w	p	w	o	e	n	a	g		e
96	x	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	s	g		e
97	f	y	n	t	l	f	c	b	p	e	r	s	y	w	w	p	w	o	p	n	s	g		e
98	x	s	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	n	g		e
99	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	s	g		e
100	x	y	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	g		e

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	

Optical Character Recognition

- Task: Learn to discriminate between (binary) images of handwritten digits
- Application: ZIP code recognition, license plate recognition, automatic check reading, etc.



Discriminant Functions

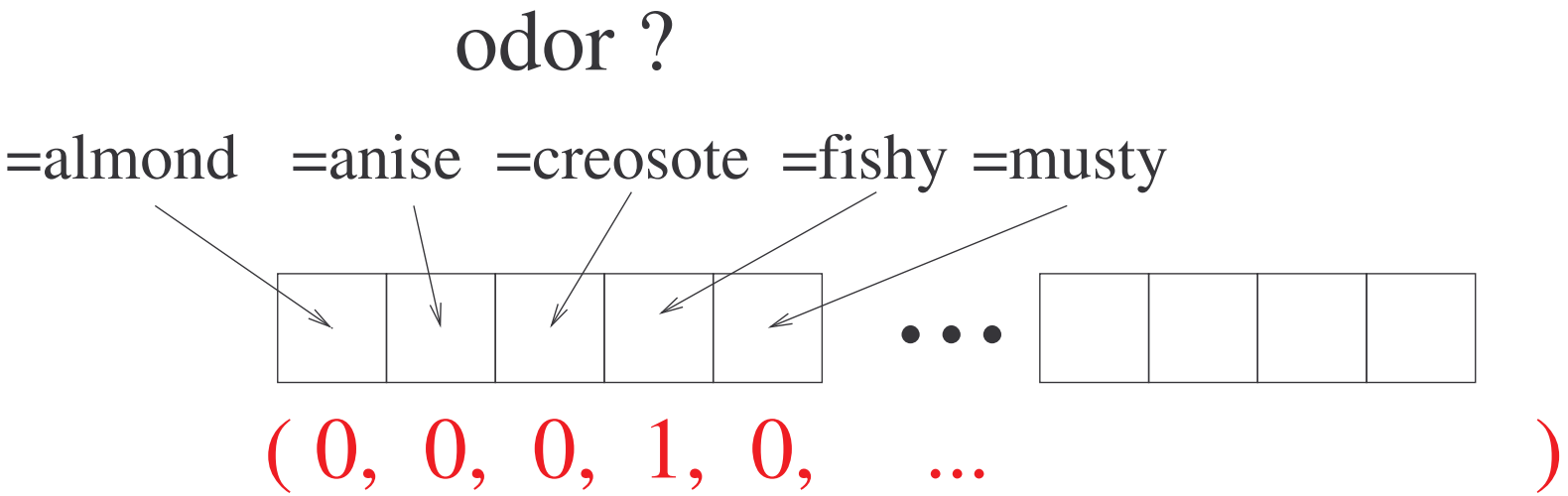
- How can all these problems be formalized?
- Step 1:
  - Come up with a numerical/binary representation of the “items” of interest (documents, mushrooms, images, etc.)
- Step 2:
  - Learn a discriminant function
  - Simplest case: instances are represented by binary attributes {0,1} and they should be classified of whether they belong to a specific class or do not

Representation: Documents

- What is a suitable (&simple) representation of documents via binary attributes?
- Word occurrence attributes: Does a specific term occur in the document or not?
  - If it does, the attribute is 1, otherwise 0.
  - Notice that the number of attributes per document equals the number of words we use (vocabulary).

Representation: Mushrooms

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



Representation: Mushrooms

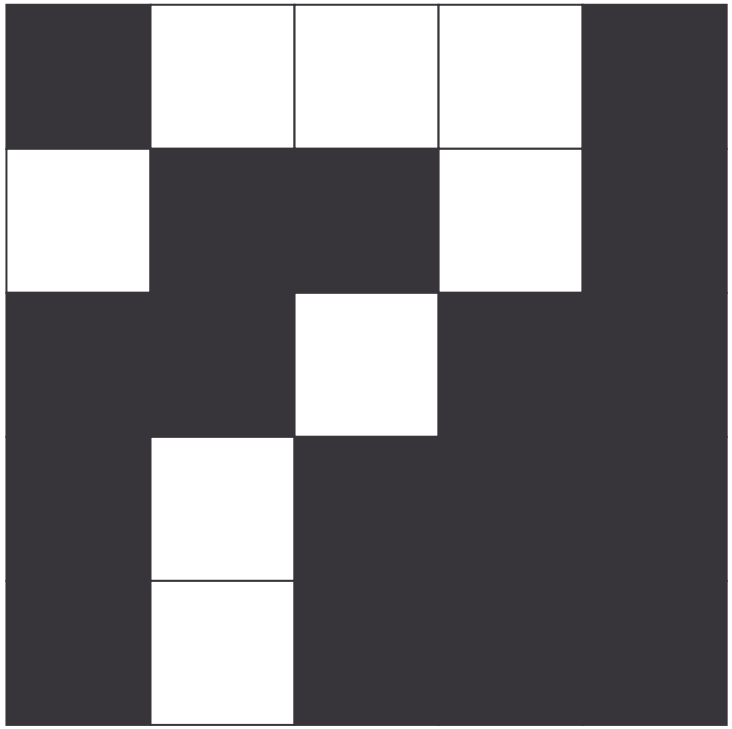
Concatenated orthogonal encoding

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

Representation: Binary Images

- Simplest case: binary images



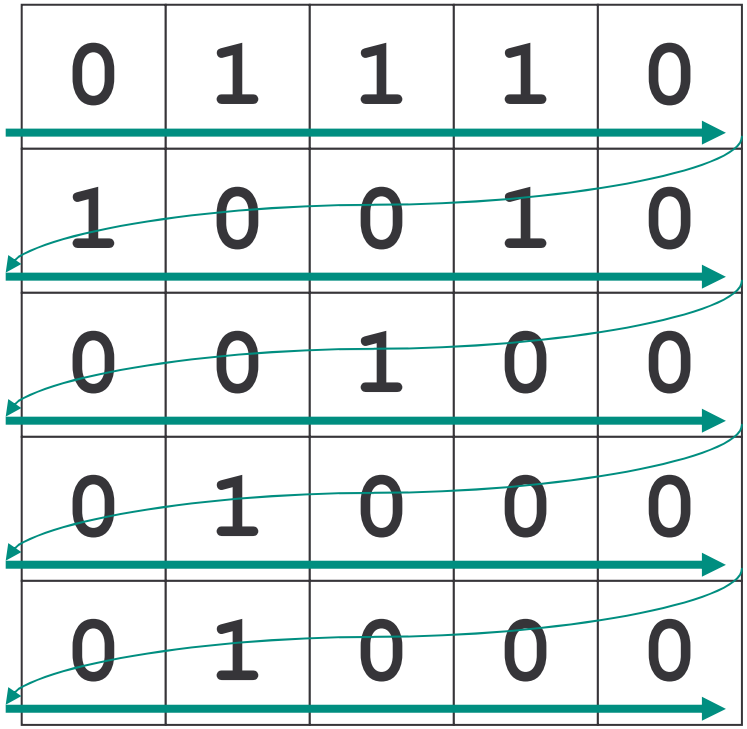


Image scan: `0111010010001001000010000`

Representation: Patterns

- Hence the following representation might be useful for individual binary patterns
- (we will not use private members, but declare everything public to keep things simple...)

```
#define NUM_ATTRIBUTES 100
class Pattern {
public:
  Pattern() { label = false; }
  bool attributes [NUM_ATTRIBUTES];
  bool label;
} ;
```

Representation: Pattern Set

- A set of patterns (e.g. for training can be implemented in a separate class)
- We use a dynamic memory management:

```
class PatternSet {
public:
  PatternSet(int num) { patterns = new Pattern[num];}
  ~PatternSet() { delete [] patterns; }
  Pattern *patterns;
} ;
```

Destructor: is called when object ceases to exist

delete *and* delete [] *frees space allocated by* new *and* new []

A Load & Save Function

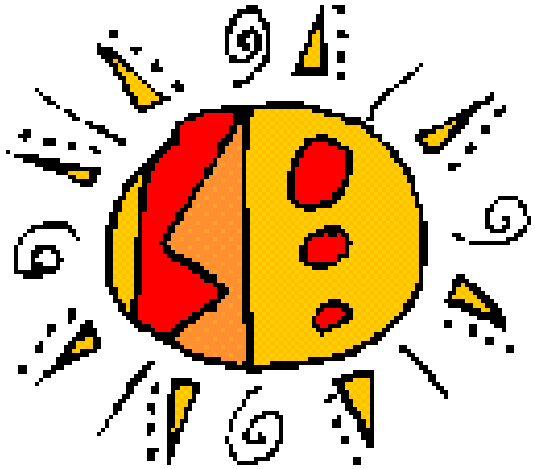
- Data can be read from and written to files in binary format

```
class PatternSet {
public:
[...]
  void read  (const string& fileName) ;
  void write (const string& fileName) const;
} ;
```

In the afternoon lab, you don't have to worry about this, since the read & write methods have been implemented for you.

break

- It’s time for a break – 10 minutes!



Linear Discriminant

- Linear discriminant functions

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

- Each attribute which is present contributes with a weight that may be positive or negative.
- A positive weight indicates that the attribute provides evidence in favor of the specific class.
- 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.
- Learning problem has been restricted to a pre-specified hypothesis class (linear discriminants)
- Learning procedure = parameter fitting

Perceptron Algorithm

Algorithm for adjusting weights

- 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

- How exactly are weights and bias adjusted?
- If the prediction is “negative”, but the example is “positive”, then
  - Add pattern to the weight vector
  - Increment the bias by one
- If the prediction is “positive”, but the example is “negative”, then
  - Subtract pattern from the weight vector
  - Decrement bias by one

Perceptron: Example

- Say the weight vector and bias are
  - w= (10, -5, 7, -1), b=-4
- A training pattern
  - x=(1,1,0,0) which is a *negative* example
- Compute output
  - f(x) = 1*10 + 1*(-5) + 0*7 + 0*(-1) – 4 = 1 > 0
  - Answer is *positive*, which is incorrect
- Update
  - w = w - x , i.e. w=(9,-6,7,-1), b = b - 1, i.e. b=-5
  - (new output would be f(x) = 9 – 6 - 5 = -1 < 0 !)

Perceptron: Theory

- It can be shown that the perceptron algorithm converges (zero mistakes), if the training patterns are linearly separable
  - *Which means, a linear discriminant function with perfect separation exists.*
- In general, it may not converge, but still yield an "appropriate" solution

- Remark: Much better algorithms have been developed in Machine Learning over the last decade! Computers now achieve human level performance in OCR.

Perceptron Lab

- In the afternoon lab, you will implement the perceptron algorithm.
- The following classes have already been implemented for you:

```
#define NUM_ATTRIBUTES 784

class Pattern {
public:
   Pattern() { label = false; }
   bool attributes[NUM_ATTRIBUTES];
   bool label;
   void load(istream& is);
   void save(ostream& os) const ;
   void print(int ncol=28) const ;
};
```

Binary images are 28 x 28.

Perceptron Lab

- continued

```
class PatternSet {
public:
    PatternSet(int n);
    ~PatternSet();
    void load(string s) ;
    void save(string s) const;
    Pattern& getPattern(int n);
    int getNum() const;

protected:
    //...
} ;
```

Perceptron Lab

- Your job is to implement the Perceptron class:

```
class Perceptron : public PatternSet {
public:
   Perceptron(int n);

   //-- perform a learning cycle using patterns from start to end
   //-- returns number of updates performed
   int learn(int start, int end);

   //-- compute number of errors on patterns from start to end
   int error(int start, int end);

   //-- print incorrect examples one by one
   void printMistakes(int start, int end) const;
protected:
   //-- use i-th pattern for update
   void update(int i) ;
   //-- make prediction for i-th pattern
   bool predict(int i) const ;

   int w[NUM_ATTRIBUTES] ; //-- weight vector
   int b;                  //-- bias (threshold)
} ;
```

The End