Introduction to Artificial Intelligence (AI)

Computer Science cpsc502, Lecture 14

Oct, 27, 2011

Slide credit: C. Conati, S. Thrun, P. Norvig, WEKA book

Today Oct 27

Machine Learning

- Introduction
- Supervised Machine Learning
  - Naïve Bayes
  - Markov-Chains (*learning the parameters of the model*)
  - Decision Trees

Machine Learning

Up until now: how to reason in a model and how to make optimal decisions

Machine learning: how to acquire a model on the basis of data / experience

- Learning parameters (e.g. probabilities)
- Learning structure (e.g. BN graphs)
- Learning hidden concepts (e.g. clustering)

Why is Machine Learning So Popular?

- **We have lots of data!**
  - Web
  - User Behavior on the Web
  - Human Genome
  - Huge medical, financial …. databases

- Need for autonomous Agents **(robots and soft-bots)** that cannot be pre-programmed

Fielded applications

- The result of learning—or the learning method itself—is deployed in practical applications
  - Processing loan applications
  - Screening images for oil slicks
  - Electricity supply forecasting
  - Diagnosis of machine faults
  - Marketing and sales
  - Separating crude oil and natural gas
  - Reducing banding in rotogravure printing
  - Finding appropriate technicians for telephone faults
  - Scientific applications: biology, astronomy, chemistry
  - Automatic selection of TV programs
  - Monitoring intensive care patients

More details on some of these apps at the end

Machine Learning

Semi sup

Supervised Learning

- Examples of correct answers are given
  - Discrete answers: **Classification**

    Category of a document, User Type, ….
  - Continuous answers: **Regression**

    Stock Price, Time of next Earthquake

data

Unsupervised Learning

- No feedback from teacher; detect patterns

Reinforcement Learning

- Feedback consists of rewards/punishment (Robotics, Interactive Systems)

Agents

Example Classification Data

	Action	Author	Thread	Length	Where
e1	skips	known	new	long	home
e2	reads	unknown	new	short	work
e3	skips	unknown	old	long	work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

k o s h ?

We want to classify new examples on property ***Action*** based on the examples' *Author, Thread, Length*, and *Where.*

Supervised ML: Formal Specification

ONE EXAMPLE

$$X_{11} \quad X_{12} \quad \cdots \quad X_{1N} \rightarrow Y_1$$

$$X_{21} \quad \cdots \quad X_{2N} \rightarrow Y_2$$

$$\vdots$$

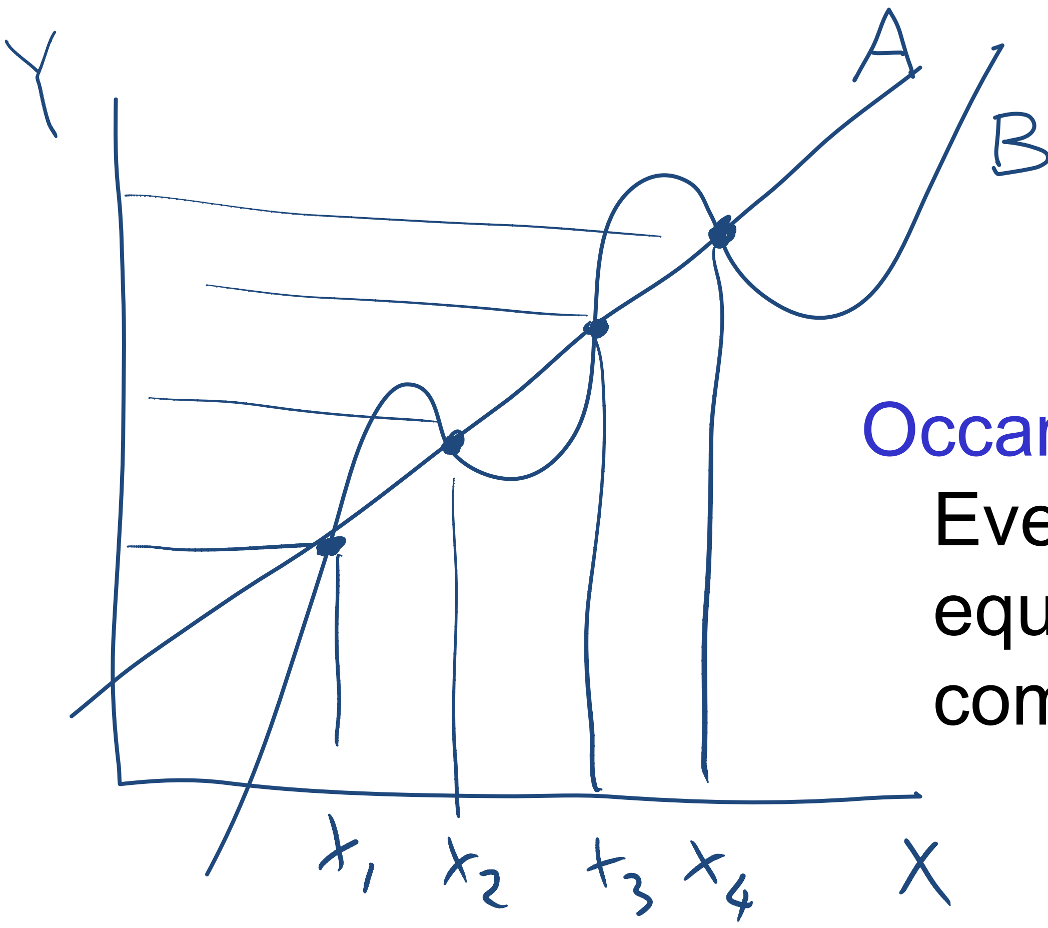
$$X_{M1} \quad \cdots \quad X_{MN} \rightarrow Y_M$$

N features

M examples

$$\boxed{f(\bar{X})} \xrightarrow{?} Y$$

The ML critical question

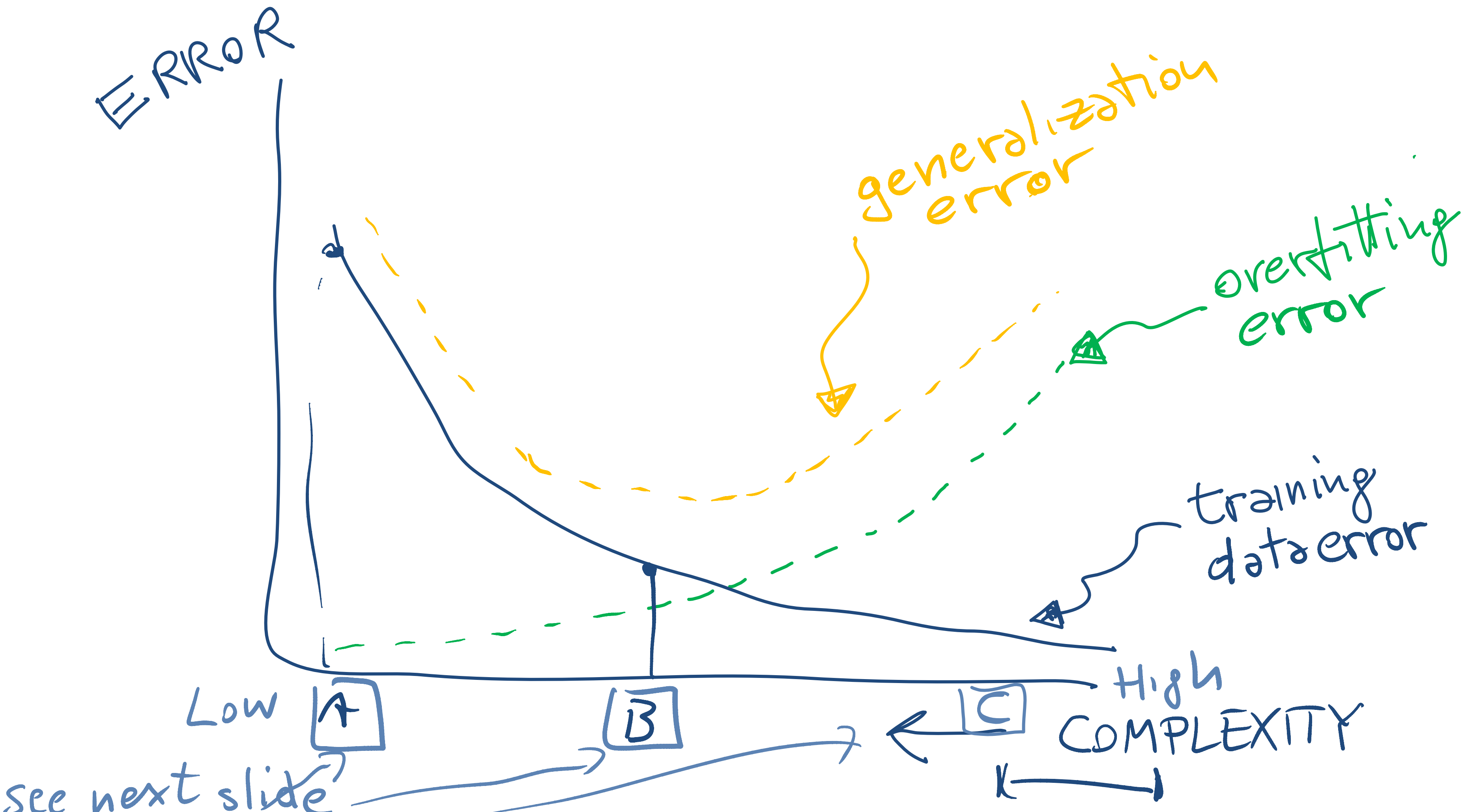


Occam's Razor:

Everything else being equal choose the less complex Hypothesis

A better, but it is not so simple...

Key trade-off between FIT and COMPLEXITY

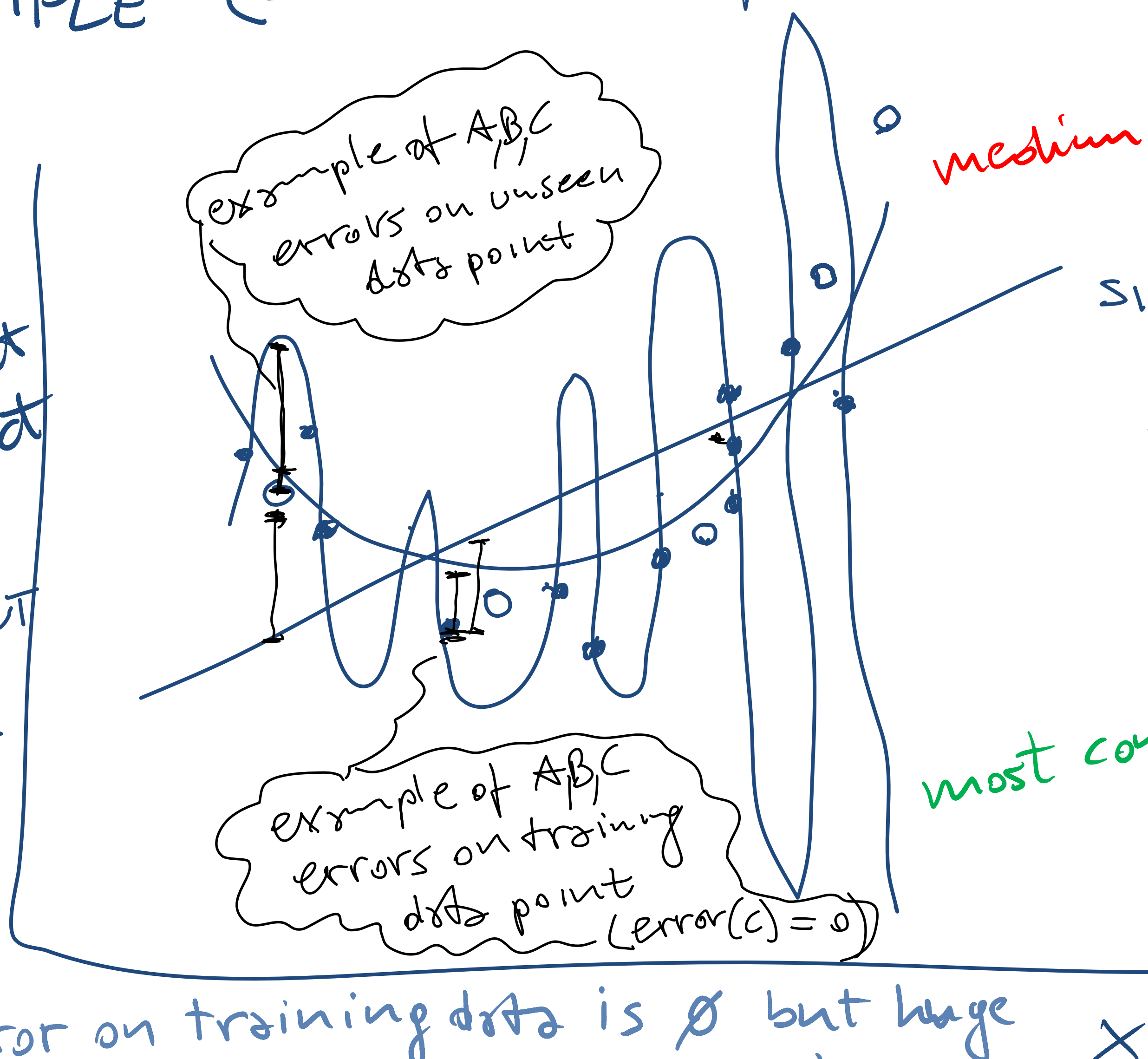


EXAMPLE (colors unrelated to previous slide)

Y

label we want to predict

WHAT ABOUT $\boxed{A}$ and $\boxed{B}$



$\boxed{B}$ medium complexity

Simpler Hypothesis $\boxed{A}$

most complex $\boxed{C}$

training data

unseen testing data

$\boxed{C}$ error on training data is $\emptyset$ but huge on unseen data - So it is not a good generalization

Today Oct 27

Machine Learning Intro

- Definitions
- Supervised Machine Learning
  - **Naïve Bayes**
  - Markov-Chains
  - Decision Trees

Naïve Bayesian Classifier

A very simple and successful Bnets that allow to classify entities in a set of classes C, given a set of attributes

Example:

- Determine whether an email is spam (only two classes spam=T and spam=F)
- Useful attributes of an email ? words contained in the email

Assumptions

- The value of each attribute depends on the classification
- **(Naïve)** The attributes are independent of each other given the classification

P("bank" | "account" , spam=T) ≠ P("bank" | spam=T)

Naïve Bayesian Classifier for Email Spam

Assumptions

- The value of each attribute depends on the classification
- **(Naïve)** The attributes are independent of each other given the classification

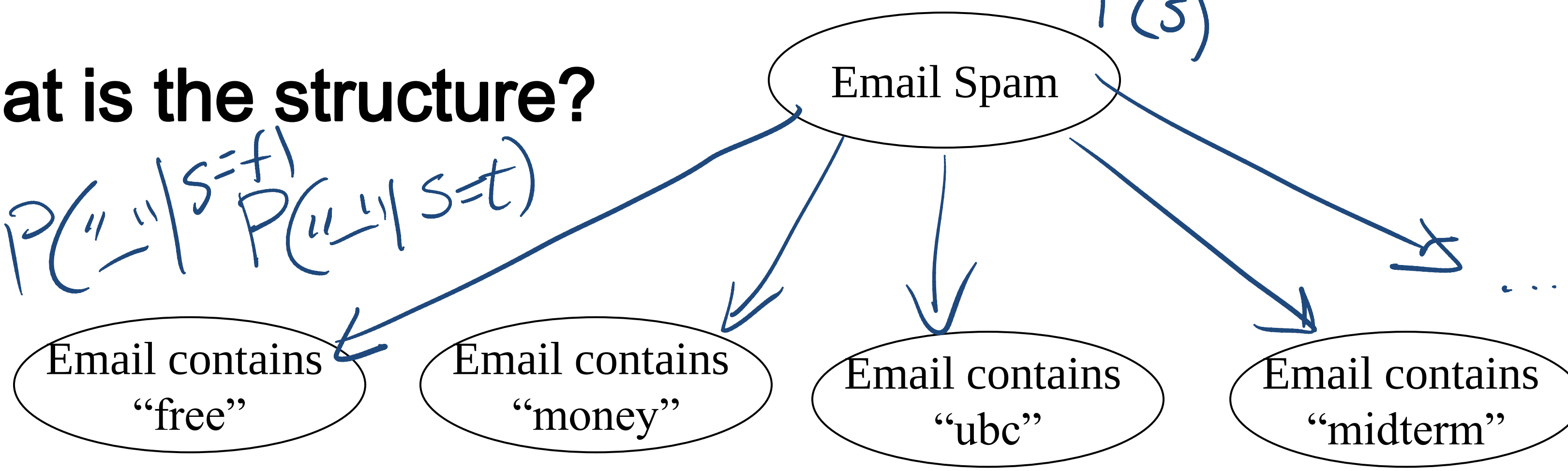
- **What is the structure?**

$P(S)$

$P(\text{"\_"} \mid s=f)$ $P(\text{"\_"} \mid s=t)$

words

10^5



Number of parameters? $2+2*10^5$

Easy to acquire?

If you have a large collection of emails for which you know if they are spam or not……

Learn the Probabilities

- You have 100,000 emails of which 10,000 are spam

$P(spam=T) = .1$ $P(spam=F) = .9$

- 2,000 of you non-spam emails contain the word "money". In contrast, "money" appears in 2,500 of your spam emails

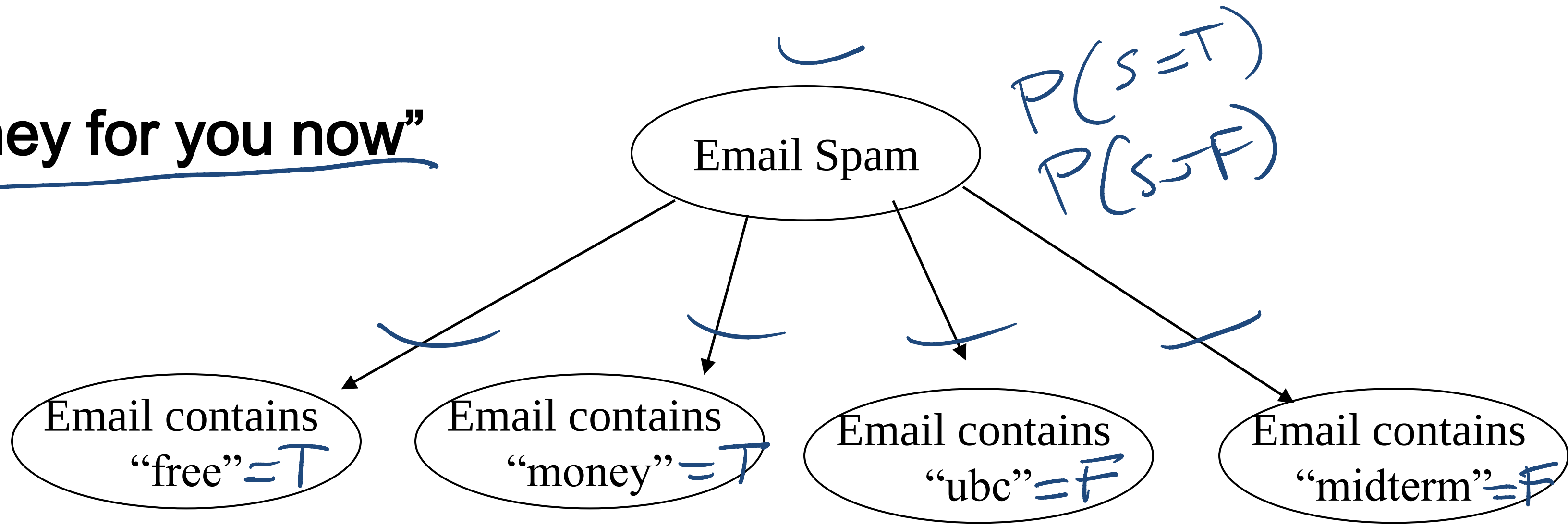
$P(\text{"money"} \mid spam=F) = 2/90$ $P(\text{"money"} \mid spam=T) = 1/4$

NB Classifier for Email Spam: Usage

Most likely class given set of observations

Is a given Email *E* spam?

"free money for you now"



Email is a spam if……. $P(S=T) > P(S=F)$

after the two probs are updated in light of the evidence (words in email are set to T)

How to improve this model?

Need more features– words aren’t enough!

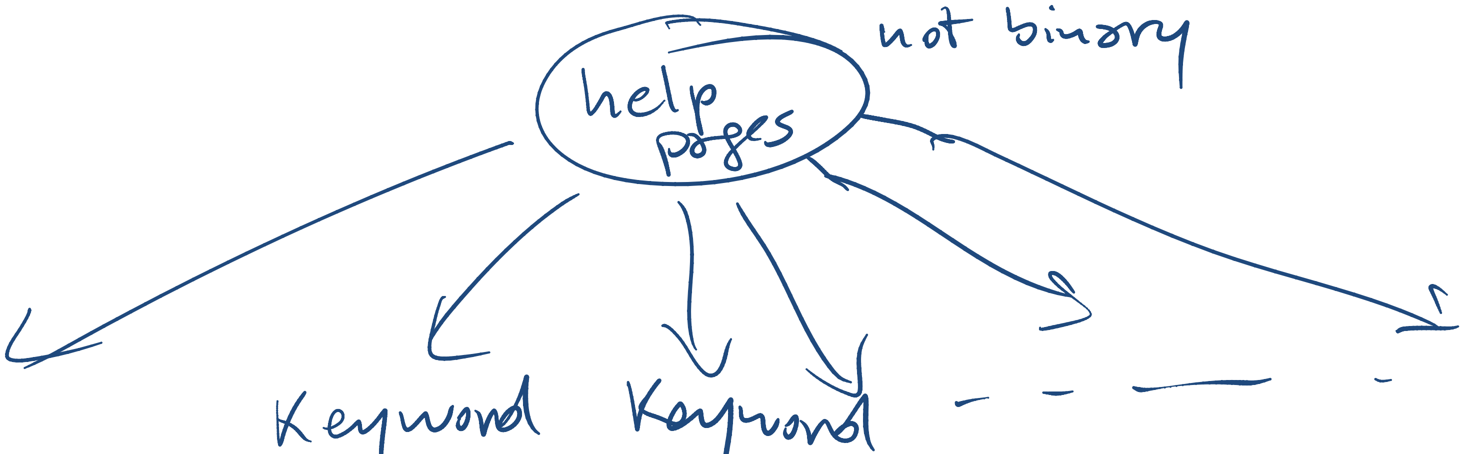
- Have you emailed the sender before?
- Have 1K other people just gotten the same email?
- Is the sending information consistent?
- Is the email in ALL CAPS?
- Do inline URLs point where they say they point?
- Does the email address you by (your) name?

Can add these information sources as new variables in the Naïve Bayes model

For another example of naïve Bayesian Classifier

See textbook ex. 6.16 (Section 6.3.1)

help system to determine what **help page a user is interested in** based on **the keywords they give in a query** to a help system.



Today Oct 27

Machine Learning

- Introduction
- Supervised Machine Learning
  - Naïve Bayes
  - **Markov-Chains**

  (learning the parameters of the model)
  - Decision Trees

Key problems in NLP

Noun Verb

"Book me a room near UBC"

w_1 w_2 w_3 w_4 w_5 w_6

$$\boxed{P(w_1, .., w_n)?}$$

Assign a probability to a sentence (a sequence of words)

- Part-of-speech tagging
- Word-sense disambiguation,
- Probabilistic Parsing

Summarization, Machine Translation…….

Predict the next word

$$P(w_n \mid w_1 \ldots w_{n-1}) = P(w_1 \ldots w_n) / P(w_1 \ldots w_{n-1})$$

- Speech recognition
- Hand-writing recognition
- Augmentative communication for the disabled

$$\boxed{P(w_1, .., w_n)?}$$

Impossible to estimate ☹

Prob of a sentence: N-Grams

Chain-rule

$$P(w_1,..,w_n) = P(w_1^n) = P(w_1)\prod_{k=2}^{n} P(w_k \mid w_1^{k-1})$$

simplifications

$$P(w_1,..,w_n) = P(w_1^n) = P(w_1)\prod_{k=2}^{n} P(w_k)$$ *unigram*

$$P(w_1,..,w_n) = P(w_1^n) = P(w_1)\prod_{k=2}^{n} P(w_k \mid w_{k-1})$$ *bigram*

$$P(w_1,..,w_n) = P(w_1^n) = P(w_1)\prod_{k=2}^{n} P(w_k \mid w_{k-1}, w_{k-2})$$ *trigram*

Bigram
<s>The big red dog barks

$$P(w_1,..,w_n) = P(w_1^n) = P(w_1 | < S >) \prod_{k=2}^{n} P(w_k | w_{k-1})$$

P(The big red dog barks)=

P(The|<S>) x P(big|The) x P(red|big) .....

Estimates for Bigrams

count

$$P(w_n \mid w_{n-1}) = \frac{P(w_{n-1}, w_n)}{P(w_{n-1})} = \frac{\dfrac{C(w_{n-1}, w_n)}{\cancel{N_{pairs}}}}{\dfrac{C(w_{n-1})}{\cancel{N_{words}}}} = \boxed{\frac{C(w_{n-1}, w_n)}{C(w_{n-1})}}$$

of consecutive words

e.g <s> w_1 w_2 w_3 w_4

4 words

4 pairs

<s>w_1
$w_1 w_2$
$w_2 w_3$
$w_3 w_4$

Estimates for Bigrams

$P(w_i \mid w_{i-1})$

Silly language repositories with **only two sentences**:

"<S> *The big red dog barks against ~~the big pink dog~~*"

"<S> *~~The big pink dog~~ is much smaller*"

17 tokens

9 types

$$P(red \mid big) = \frac{P(big, red)}{P(big)} = \frac{\dfrac{C(big, red)}{\cancel{N_{pairs}}}}{\dfrac{C(big)}{\cancel{N_{words}}}} = \frac{C(big, red)}{C(big)} = \frac{1}{3}$$

count

Count how many times in your documents you have "big red" and "big"

$P(w_i \mid w_{i-1})$

$10^5 * 10^5$ matrix

$$\boxed{P(w_i \mid w_{i-1}, w_{i-2})}$$

some models use two preceeding words

Berkeley Restaurant Project (1994)
Table: Counts *Dialog System*

Corpus: ~10,000 sentences, 1616 word types

w_n 10^5 word tokens

w_{n-1}

	I	want	to	eat	Chinese	food	lunch
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

Excerpt of a 1616x1616 table

$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

BERP Table: $P(w_n \mid w_{n-1})$

w_n

w_{n-1}

	I	want	to	eat	Chinese	food	lunch
I	.0023	.32	0	.0038	0	0	0
want	.0025	0	.65	0	.0049	.0066	.0049
to	.00092	0	.0031	.26	.00092	0	.0037
eat	0	0	.0021	0	.020	.0021	.055
Chinese	.0094	0	0	0	0	.56	.0047
food	.013	0	.011	0	0	0	0
lunch	.0087	0	0	0	0	.0022	0

BERP Table Comparison

$\boxed{w_n}$

$\boxed{w_{n-1}}$

Counts

	I	want	to	eat	Chinese	food	lunch
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

Prob.

$$\frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

	I	want	to	eat	Chinese	food	lunch
I	.0023	.32	0	.0038	0	0	0
want	.0025	0	.65	0	.0049	.0066	.0049
to	.00092	0	.0031	.26	.00092	0	.0037
eat	0	0	.0021	0	.020	.0021	.055
Chinese	.0094	0	0	0	0	.56	.0047
food	.013	0	.011	0	0	0	0
lunch	.0087	0	0	0	0	.0022	0

1?

Two problems with applying:

$$P(w_1^n) = P(w_1)\prod_{k=2}^{n} P(w_k \mid w_{k-1})$$

to

$\boxed{w_n}$

<s>

$\boxed{w_{n-1}}$

	I	want	to	eat	Chinese	food	lunch
I	.0023	.32	0	.0038	0	0	0
want	.0025	0	.65	0	.0049	.0066	.0049
to	.00092	0	.0031	.26	.00092	0	.0037
eat	0	0	.0021	0	.020	.0021	.055
Chinese	.0094	0	0	0	0	.56	.0047
food	.013	0	.011	0	0	0	0
lunch	.0087	0	0	0	0	.0022	0

General Problems for ML !

Problem (1)

We may need to multiply many very small numbers (underflows!)

Easy Solution:

- Convert probabilities to logs and then sum
- To get the real probability (if you need it) go back to the antilog.

Problem (2)

The probability matrix for n-grams is sparse

How can we assign a probability to a sequence where one of the component n-grams has a value of zero?

Solutions:

- Add-one smoothing (Laplace Smoothing)
- *Good-Turing*
- *Back off and Deleted Interpolation*

Add-One

Make the zero counts 1.

Rationale: If you had seen these "rare" events chances are you would only have seen them once.

NEXT PAGE

Corpus: ~10,000 sentences, 1616 word types V

$N = \sim 10^5$ word tokens

show $\sum_{w \in V} P^*(w) = 1$

unigram

How many times word w appears in my data

count

$$P(w) = \frac{C(w)}{N} \longrightarrow \boxed{P^*(w)} = \frac{C(w)+1}{N+|V|}$$

$$P^*(w) = \frac{C^*(w)}{N} \longrightarrow C^*(w) = (C(w)+1)\frac{N}{N+|V|}$$

Pseudo-counts

$$\sum_{w \in V} P^*(w) = \sum_{w \in V} \frac{C(w)+1}{N+|V|} =$$

all the word tokens

$$= \frac{\sum_{w \in V} C(w) + 1}{N+|V|} = \frac{\sum_{w \in V} C(w) + \sum_{w \in V} 1}{N+|V|} =$$

N and |V| don't depend on w

Sum up 1 as many times as there are elements in V

$$= \frac{N+|V|}{N+|V|} = 1$$

Add-One: Bigram

$\boxed{w_n}$

$\boxed{w_{n-1}}$

Counts

	I	want	to	eat	Chinese	food	lunch
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})} \longrightarrow P^*(w_n \mid w_{n-1}) = \frac{C(w_{n-1}, w_n)+1}{C(w_{n-1})+V}$$

…… $\longrightarrow$

$$C^*(w_{n-1}, w_n) = (C(w_{n-1}, w_n)+1)\frac{N}{C(w_{n-1})+V}$$

BERP Original vs. Add-one smoothed Counts

$\boxed{w_n}$

$\boxed{w_{n-1}}$

	I	want	to	eat	Chinese	food	lunch
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

	I	want	to	eat	Chinese	food	lunch
I	6	740	.68	10	.68	.68	.68
want	2	.42	331	.42	**5**	4	3
to	**2**	.69	8	594	**2**	.69	9
eat	.37	.37	1	.37	**15**	1	20
Chinese	.36	.12	.12	.12	.12	15	.24
food	10	.48	9	.48	.48	.48	.48
lunch	1.1	.22	.22	.22	.22	.44	.22

Biggest Language Model…

Google language model
Update (22 Sept. 2006): The LDC has the data available in their catalog. The counts are as follows:
File sizes: approx. 24 GB compressed (gzip'ed) text files

Number of tokens: 1,024,908,267,229 $= N$
Number of sentences: 95,119,665,584
Number of unigrams: 13,588,391 $= V$
Number of bigrams: 314,843,401 $\leftarrow w_1 w_2$
Number of trigrams: 977,069,902
Number of fourgrams: 1,313,818,354

Today Oct 27

<u>Machine Learning</u>

- Introduction
- Supervised Machine Learning
  - Naïve Bayes
  - Markov Chains
  - **Decision Trees**

Example Classification Data

	Action	Author	Thread	Length	Where
e1	skips	known	new	long	home
e2	reads	unknown	new	short	work
e3	skips	unknown	old	long	work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

We want to classify new examples on property ***Action*** based on the examples' *Author, Thread, Length*, and *Where.*

Learning task

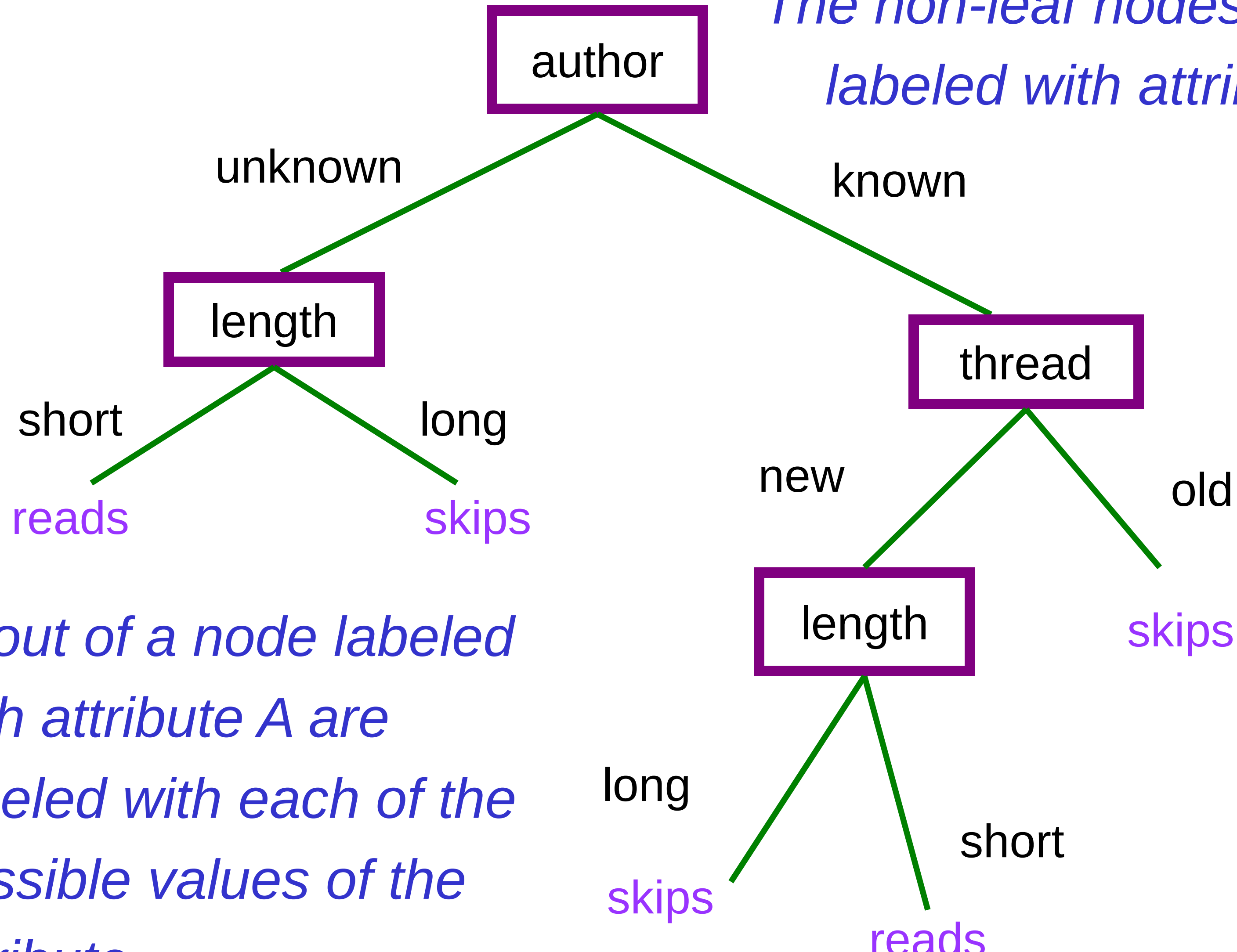
- **Inductive inference**
  - Given a set of examples of

    f(author,thread, length, where) = {reads,skips}

  - Find a function *h(author,thread, length, where)* that approximates *f*

Example Decision Tree

The non-leaf nodes are labeled with attributes.



Arcs out of a node labeled with attribute A are labeled with each of the possible values of the attribute

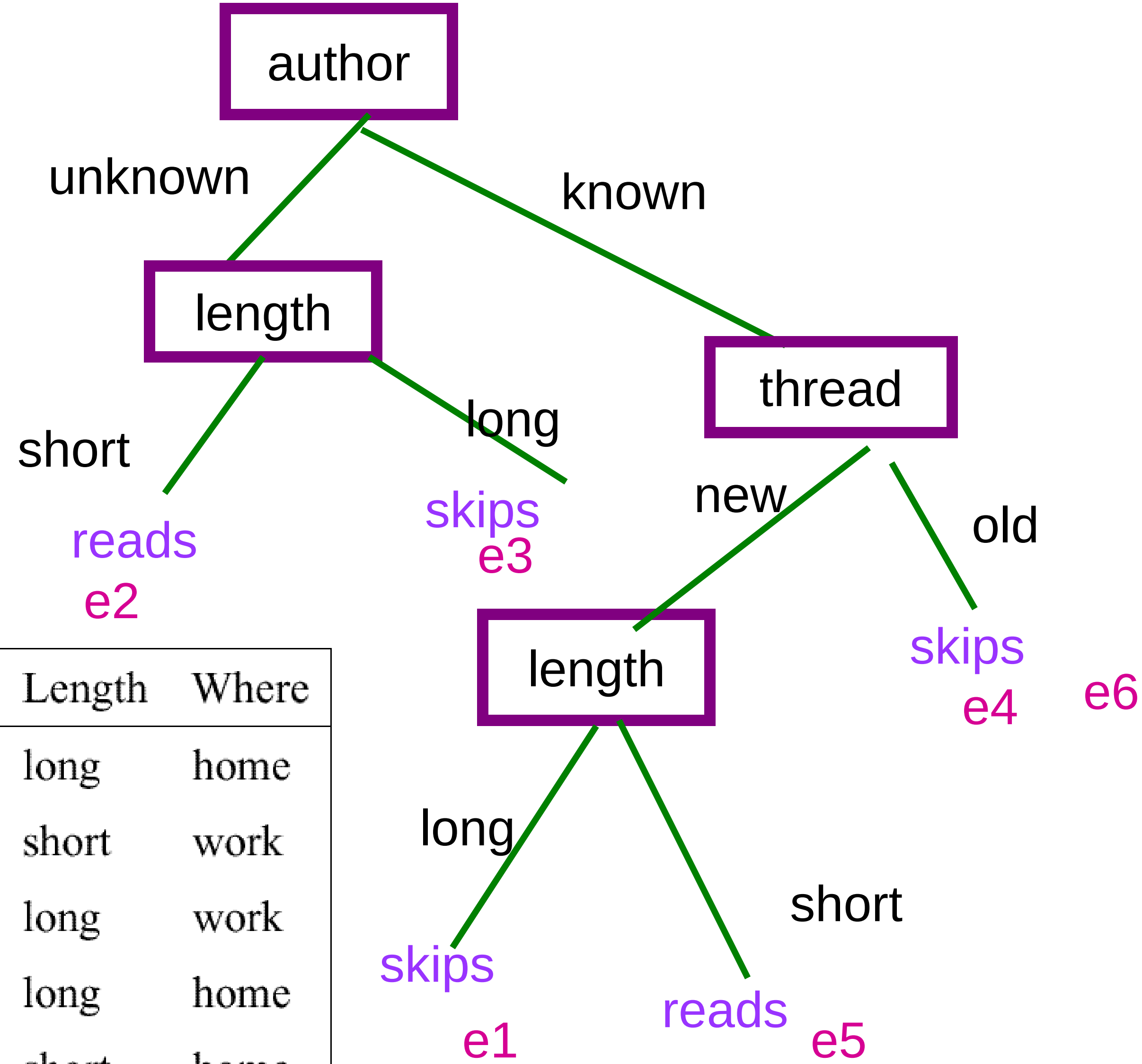
The leaves of the tree are labeled with classifications.

DT as classifiers

To classify an example, filter in down the tree

- For each attribute of the example, follow the branch corresponding to that attribute's value.
- When a leaf is reached, the example is classified as the label for that leaf.

DT as classifiers



	Action	Author	Thread	Length	Where
e1	skips	known	new	long	home
e2	reads	unknown	new	short	work
e3	skips	unknown	old	long	work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

Learning Decision Trees

Method for supervised classification (we will assume attributes with finite discrete values)

- Representation is a **decision tree**.
- Bias is towards simple decision trees.
- Search through the space of decision trees, from simple decision trees to more complex ones.

DT Applications

- **DT are often the first method tried in many areas of industry and commerce**, when task involves learning from a data set of examples
- Main reason: **the output is easy to interpret by humans**

Equivalent Rule Based Representation

If *author* is *unknown* and *length* is *short*
then *user-action* is *reads*

If *author* is *unknown* and *length* is *long*
then *user-action* is *skips*

If *author* is *known* and *thread* is *new* and *length* is *short*
then *user-action* is *reads*

If *author* is *known* and *thread* is *new* and *length* is *long*
then *user-action* is *skips*

If *author* is *known* and *thread* is *old*
then *user-action* is *skips*
Suppose this is the true criteria that my user is employing

TODO for next Tue

- **Read textbook 7.3**
- Also Do exercise 7.A

http://www.aispace.org/exercises.shtml

Processing loan applications
(American Express)

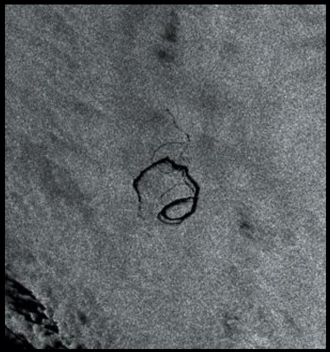
- Given: questionnaire with financial and personal information
- Question: should money be lent?
- Simple statistical method covers 90% of cases
- Borderline cases referred to loan officers
- But: 50% of accepted borderline cases defaulted!
- Solution: reject all borderline cases?
  - No! Borderline cases are most active customers

Enter machine learning

- 1000 training examples of borderline cases
- 20 attributes:
  - age
  - years with current employer
  - years at current address
  - years with the bank
  - other credit cards possessed,…
- Learned rules: correct on 70% of cases
  - human experts only 50%
- Rules could be used to explain decisions to customers

Screening images

- Given: radar satellite images of coastal waters
- Problem: detect oil slicks in those images
- Oil slicks appear as dark regions with changing size and shape
- Not easy: lookalike dark regions can be caused by weather conditions (e.g. high wind)
- Expensive process requiring highly trained personnel


Enter machine learning

- Extract dark regions from normalized image
- Attributes:
  - size of region
  - shape, area
  - intensity
  - sharpness and jaggedness of boundaries
  - proximity of other regions
  - info about background
- Constraints:
  - Few training examples—oil slicks are rare!
  - Unbalanced data: most dark regions aren't slicks
  - Regions from same image form a batch
  - Requirement: adjustable false-alarm rate

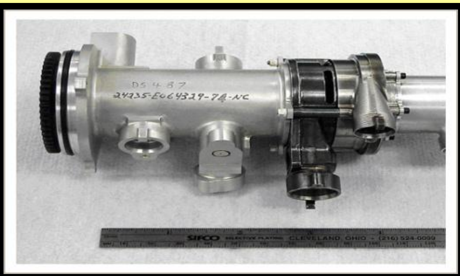
Load forecasting

- Electricity supply companies need forecast of future demand for power
- Forecasts of min/max load for each hour ® significant savings
- Given: manually constructed load model that assumes “normal” climatic conditions
- Problem: adjust for weather conditions
- Static model consist of:
  - base load for the year
  - load periodicity over the year
  - effect of holidays

Enter machine learning

- Prediction corrected using "most similar" days
- Attributes:
  - temperature
  - humidity
  - wind speed
  - cloud cover readings
  - plus difference between actual load and predicted load
- Average difference among three "most similar" days added to static model
- Linear regression coefficients form attribute weights in similarity function

Diagnosis of machine faults



- Diagnosis: classical domain of expert systems
- Given: Fourier analysis of vibrations measured at various points of a device's mounting
- Question: which fault is present?
- Preventative maintenance of electromechanical motors and generators
- Information very noisy
- So far: diagnosis by expert/hand-crafted rules

Enter machine learning

- Available: 600 faults with expert's diagnosis
- ~300 unsatisfactory, rest used for training
- Attributes augmented by intermediate concepts that embodied causal domain knowledge
- Expert not satisfied with initial rules because they did not relate to his domain knowledge
- Further background knowledge resulted in more complex rules that were satisfactory
- Learned rules outperformed hand-crafted ones

Marketing and sales I

- Companies precisely record massive amounts of marketing and sales data
- Applications:
  - Customer loyalty:
    identifying customers that are likely to defect by detecting changes in their behavior
    (e.g. banks/phone companies)
  - Special offers:
    identifying profitable customers
    (e.g. reliable owners of credit cards that need extra money during the holiday season)

Marketing and sales II

- Market basket analysis
  - Association techniques find groups of items that tend to occur together in a transaction
    (used to analyze checkout data)
- Historical analysis of purchasing patterns
- Identifying prospective customers
  - Focusing promotional mailouts
    (targeted campaigns are cheaper than mass-marketed ones)

Machine learning and statistics

- Historical difference (grossly oversimplified):
  - Statistics: testing hypotheses
  - Machine learning: finding the right hypothesis
- But: huge overlap
  - Decision trees (C4.5 and CART)
  - Nearest-neighbor methods
- Today: perspectives have converged
  - Most ML algorithms employ statistical techniques