CSC 550: Introduction to Artificial Intelligence

Fall 2008

Machine learning: decision trees

- decision trees
- user-directed learning
- data mining & decision trees
  - ID3 algorithm
  - information theory
- information bias
- extensions to ID3
  - C4.5, C5.0
- further reading

Philosophical question

the following code can deduce new facts from existing facts & rules

- is this machine learning?

```
(define KNOWLEDGE
  '((itRains <-- ) (isCold <-- ) (isCold <-- itSnows)
    (getSick <-- isCold getWet) (getWet <-- itRains)
    (hospitalize <-- getSick highFever)))


(define (deduce goal known)

  (define (deduce-any goal-lists)
    (cond ((null? goal-lists) #f)
          ((null? (car goal-lists)) #t)
          (else (deduce-any (append (extend (car goal-lists) known)
                                    (cdr goal-lists))))))

  (define (extend anded-goals known-step)
    (cond ((null? known-step) '())
          ((equal? (car anded-goals) (caar known-step))
           (cons (append (cddar known-step) (cdr anded-goals))
                 (extend anded-goals (cdr known-step))))
          (else (extend anded-goals (cdr known-step)))))

  (if (list? goal)
      (deduce-any (list goal))
      (deduce-any (list (list goal)))))
```

```
> (deduce 'getSick KNOWLEDGE)
#t

> (deduce 'hospitalize KNOWLEDGE)
#f
```

in 1995, I coauthored an automated theorem proving system (SATCHMORE) that was subsequently used to solve an open-question in mathematics

is that learning?

Machine learning

machine learning: any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population.

-- Herbert Simon, 1983

clearly, being able to adapt & generalize are key to intelligence

main approaches

- symbol-based learning: the primary influence on learning is domain knowledge
  - version space search, decision trees, explanation-based learning
- connectionist learning: learning is sub-symbolic, based on brain model
  neural nets, associationist memory
- emergent learning: learning is about adaptation, based on evolutionary model
  genetic algorithms, genetic algorithms, artificial life

3

Decision trees: motivational example

recall the game "20 Questions"

1.	Is it alive	*yes*
2.	Is it an animal?	*yes*
3.	Does it fly?	*no*
4.	Does walk on 4 legs?	*no*
.		
.		
.		
10.	Does it have feathers?	*yes*

It is a penguin.

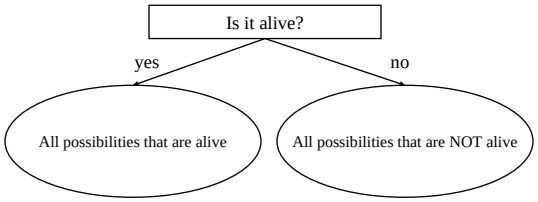
QUESTION: what is the "best" strategy for playing?

4

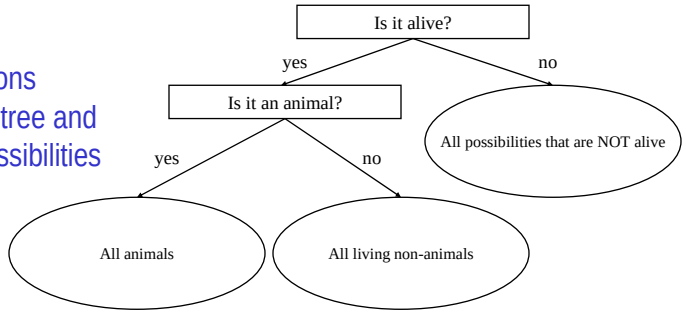
Decision trees

can think of each question as forming a branch in a search tree

- a *decision tree* is a search tree where nodes are labeled with questions and edges are labeled with answers



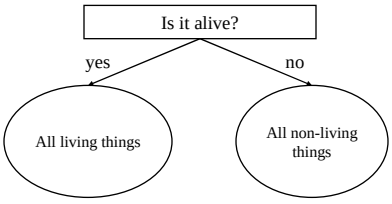
subsequent questions further expand the tree and break down the possibilities

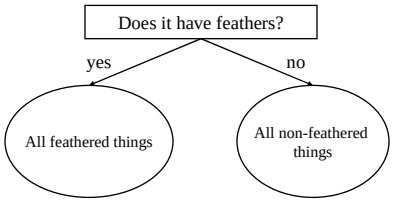


5

Decision trees

note: not all questions are created equal





ideally, want a question to divide the remaining possibilities in half

- reminiscent of binary search

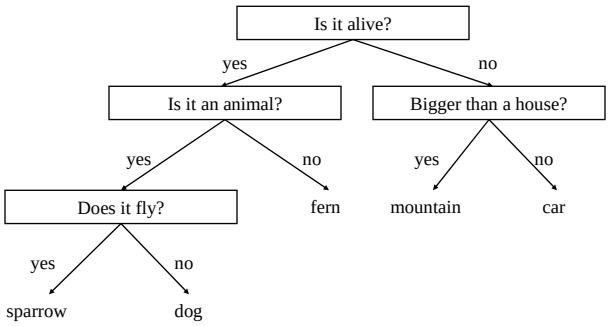
what is the maximum number of items that can be identified in 20 questions?

6

Decision trees vs. rules

decision trees can be thought of encoding rules

- traverse the edges of the trees to reach a leaf
- the path taken defines a rule



IF it is alive AND
it is an animal AND
it flies
THEN it is a sparrow.

IF it is alive AND
it is an animal AND
it does not fly
THEN it is a dog.

IF it is alive AND
it is not an animal
THEN it is a fern.

IF it is not alive AND
it is bigger than a house
THEN it is a mountain.

IF it is not alive AND
it is not bigger than a house
THEN it is a car.

7

Scheme implementation

can define a decision tree as a Scheme list

- internal nodes are questions
- left subtree is "yes", right subtree is "no"
- leaves are the things that can be identified

```
(define QUIZ-DB
  '((is it alive?)
    ((is it an animal?) dog fern)
    ((bigger than a house?) mountain car)))
```

to play the game, recursively traverse the tree, prompting the user to determine which path to take

```
(define (guess dbase)
  (if (list? dbase)
      (begin (display (car dbase))
             (if (member (read) '(y yes))
                 (guess (cadr dbase))
                 (guess (caddr dbase))))
      (begin (display "It is a ") (display dbase) (newline))))
```

8

Adding learning to the game

we could extend the game to allow for a simple kind of learning

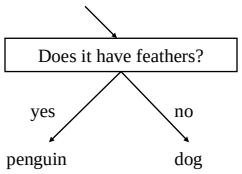
- when a leaf is reached, don't just assume it is the answer
- prompt the user – if not correct, then ask for their answer and a question that distinguishes

1. Is it alive *yes*
2. Is it an animal? *yes*
3. Does it fly? *no*
4. Is it a dog? *no*

Enter your answer: *penguin*

Enter a question that is 'yes' for penguin but 'no' for dog: *Does it have feathers?*

- then extend the tree by replacing the incorrect leaf with a new subtree



9

w/ user-directed learning

uses global variable QUIZ-DB

- load-file reads a decision tree from a file, stores in QUIZ-DB
- guess-game updates QUIZ-DB
- update-file stores the updated QUIZ-DB back in a file

```
(define QUIZ-DB 'shoe)

(define (load-file fname)
  (let ((infile (open-input-file fname)))
    (begin (set! QUIZ-DB (read infile))
           (close-input-port infile))))

(define (update-file fname)
  (let ((outfile (open-output-file fname 'replace)))
    (begin (display QUIZ-DB outfile)
           (close-output-port outfile))))

(define (guess-game)

  (define (replace-leaf dtree oldval newval)
    (cond ((list? dtree) (list (car dtree) (replace-leaf (cadr dtree) oldval newval)
                               (replace-leaf (caddr dtree) oldval newval)))
          ((equal? dtree oldval) newval)
          (else dtree)))

  (define (guess dbase)
    (if (list? dbase)
        (begin (display (car dbase)) (display " ")
               (if (member (read) '(y yes))
                   (guess (cadr dbase))
                   (guess (caddr dbase))))
        (begin (display "Is it a ") (display dbase) (display "? ")
               (if (member (read) '(y yes))
                   (begin (display "Thanks for playing!") (newline))
                   (begin (display "What is your answer? ")
                          (let ((answer (read)))
                            (begin (display "Enter a question that is true for ")
                                   (display answer) (display " (in parentheses): ")
                                   (set! QUIZ-DB (replace-leaf QUIZ-DB dbase
                                                               (list (read) answer dbase)))))))))))
  (guess QUIZ-DB))
```

10

Data mining & decision trees

decision trees can be used to extract patterns from data

- based on a collection of examples, will induce which properties lead to what

e.g., suppose we have collected stats on good and bad loans

from these examples, want to determine what properties/characteristics should guide future loans

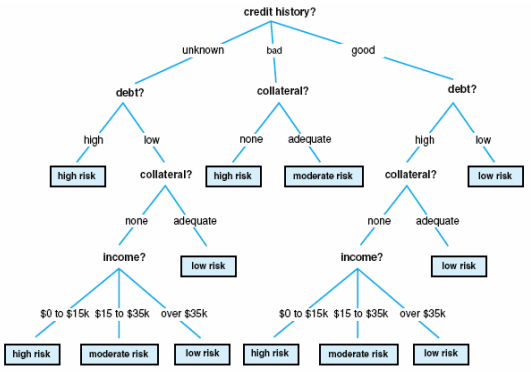
NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

11

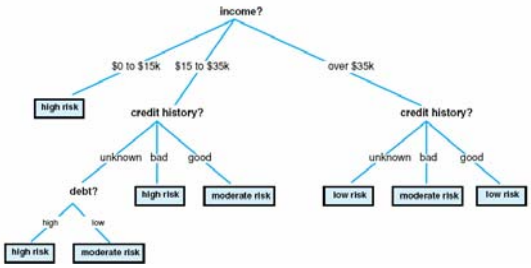
Classification via a decision tree

a decision tree could capture the knowledge in these examples

- identifies which combinations of properties lead to which outcomes



depending on which properties you focus first, you can construct very different trees



12

Generic learning algorithm

start with a population of examples, then repeatedly

- select a property/characteristic that partitions the remaining population
- add a node for that property/characteristic

more formally:

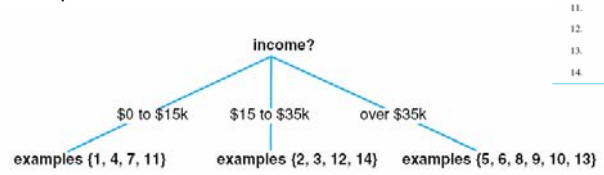
```
function induce_tree (example_set, Properties)

begin
if all entries in example_set are in the same class
   then return a leaf node labeled with that class
   else if Properties is empty
      then return leaf node labeled with disjunction of all classes in example_set
      else begin
         select a property, P, and make it the root of the current tree;
         delete P from Properties;
            for each value, V, of P,
               begin
                  create a branch of the tree labeled with V;
                  let partition_v be elements of example_set with values V for property P;
                  call induce_tree(partition_v, Properties), attach result to branch V
               end
      end
end
```


Example

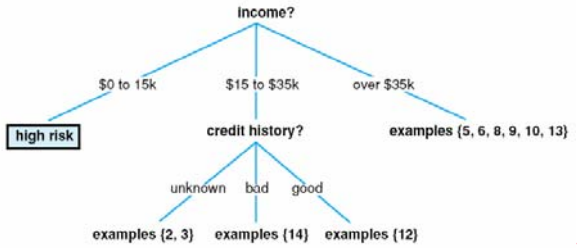
starting with the population of loans

- suppose we first select the income property
- this separates the examples into three partitions

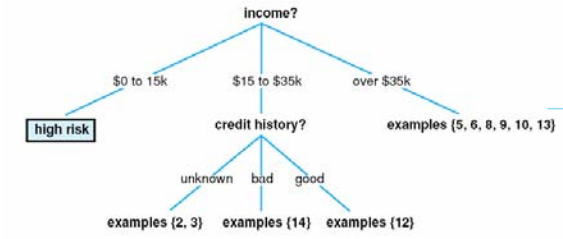
NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k



- all examples in leftmost partition have same conclusion – HIGH RISK
- other partitions can be further subdivided by selecting another property



Example (cont.)



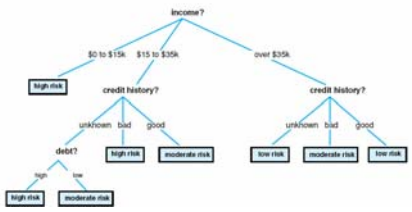
NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

15

ID3 algorithm

ideally, we would like to select properties in an order that minimizes the size of the resulting decision tree

- Occam's Razor: *always accept the simplest answer that fits the data*
- a minimal tree provides the broadest generalization of the data, distinguishing necessary properties from extraneous
  - e.g., the smaller credit risk decision tree does not even use the collateral property – not required to correctly classify all examples



the ID3 algorithm was developed by Quinlan (1986)

- a hill-climbing/greedy approach
- uses an information theory metric to select the next property
- goal is to minimize the overall tree size (but not guaranteed)

16

ID3 & information theory

the selection of which property to split on next is based on information theory

- the *information content* of a tree is defined by

$$I[tree] = \sum -prob(classification_i) * \log_2(prob(classification_i))$$

e.g., In credit risk data, there are 14 samples

prob(high risk) = 6/14
prob(moderate risk) = 3/14
prob(low risk) = 5/14

the information content of a tree that correctly classifies these examples is

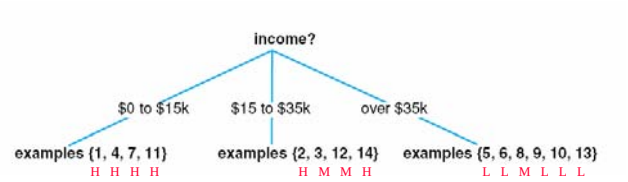
$$I[tree] = -6/14 * \log_2(6/14) + -3/14 * \log_2(3/14) + -5/14 * \log_2(5/14)$$
$$= -6/14 * -1.222 + -3/14 * -2.222 + -5/14 * -1.485$$
$$= 1.531$$

17

ID3 & more information theory

- after splitting on a property, consider the expected (or remaining) content of the subtrees

$$E[property] = \sum (\#\ in\ subtree_i\ /\ \#\ of\ samples) * I[subtree_i]$$



$$E[income] = 4/14 * I[subtree_1] + 4/14 * I[subtree_2] + 6/14 * I[subtree_3]$$
$$= 4/14 * (-4/4 \log_2(4/4) + -0/4 \log_2(0/4) + -0/4 \log_2(0/4)) +$$
$$4/14 * (-2/4 \log_2(2/4) + -2/4 \log_2(2/4) + -0/4 \log_2(0/4)) +$$
$$6/14 * (-0/6 \log_2(0/6) + -1/6 \log_2(1/6) + -5/6 \log_2(5/6))$$
$$= 4/14 * (0.0+0.0+0.0) + 4/14 * (0.5+0.5+0.0) + 6/14 * (0.0+0.43+0.22)$$
$$= 0.0 + 0.29 + 0.28$$
$$= 0.57$$

18

Credit risk example (cont.)

what about the other property options?

E[debt]? E[history]? E[collateral]?

- after further analysis
  E[income] = 0.57
  E[debt] = 1.47
  E[history] = 1.26
  E[collateral] = 1.33

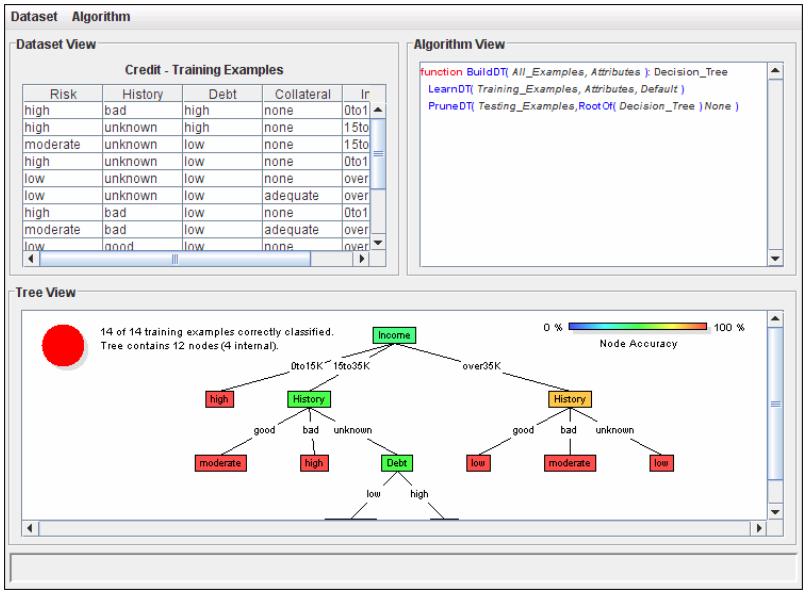
the ID3 selection rules splits on the property that produces the greatest information gain

- i.e., whose subtrees have minimal remaining content → minimal E[property]

- in this example, income will be the first property split
- then repeat the process on each subtree

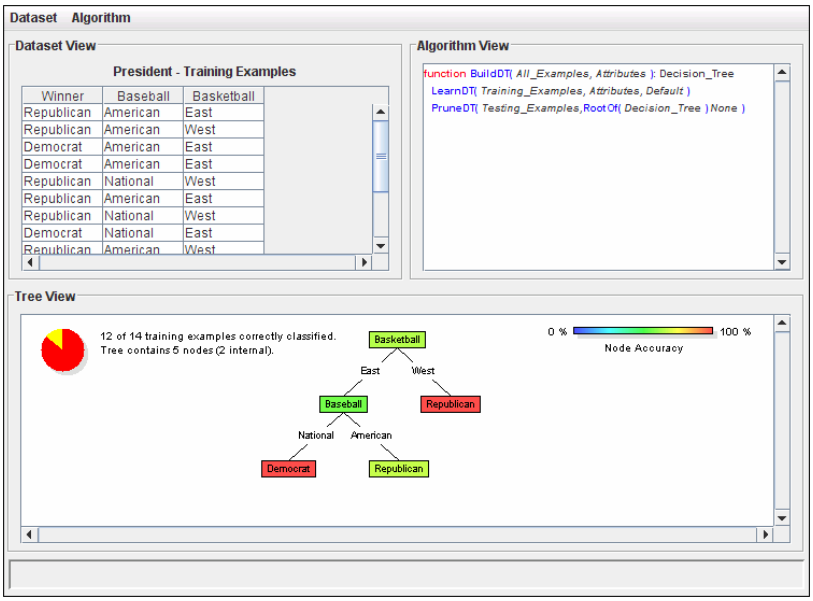
19

Decision tree applet from AIxploratorium



20

Presidential elections & sports



21

Effectiveness of ID3 in practice

Quinlan did a study of ID3 in evaluating chess boards

- limited scope to endgames involving King+Knight vs. King+Rook
- goal: recognize wins/losses within 3 moves
  search space: 1.4 million boards
- identified 23 properties that could be used by ID3

Size of Training Set	Percentage of Whole Universe	Errors in 10,000 Trials	Predicted Maximum Errors
200	0.01	199	728
1,000	0.07	33	146
5,000	0.36	8	29
25,000	1.79	6	7
125,000	8.93	2	1

22

Inductive bias

inductive bias : any criteria a learner uses to constrain the problem space

inductive bias is necessary to the workings of ID3

- a person must identify the relevant properties in the samples
- the ID3 algorithm can only select from those properties when looking for patterns
  if the person ignores an important property, then the effectiveness of ID3 is limited

technically, the selected properties must have a discrete range of values

e.g., yes, no high, moderate, low

- if the range is really continuous, it must be divided into discrete ranges

  e.g., 0to15K, 15to35K, over35K

23

Extensions to ID3

the C4.5 algorithm (Quinlan, 1993) extends ID3 to

- automatically determine appropriate ranges from continuous values
- handle samples with unknown property values
- automatically simplify the constructed tree by pruning unnecessary subtrees

the C5.0 algorithm (Quinlan, 1996) further extends C4.5 to

- be faster & make better use of memory
- produce even smaller trees by pruning more effectively
- allow for weighting the samples & better control the training process

Quinlan currently markets C5.0 and other data mining tools via his company RuleQuest Research (www.rulequest.com)

24

Further reading

Wikipedia: Data Mining

Data Mining: What is Data Mining? by Jason Frand

Can Data Mining Save America's Schools? by Marianne Kolbasuk McGee

DHS halts anti-terror data-mining program by the Associated Press

RuleQuest Research

25