

CMU SCS

15-826: Multimedia Databases and Data Mining

Data Mining - AI reminders
C. Faloutsos

CMU SCS

Outline

Goal: 'Find similar / interesting things'

- Intro to DB
- Indexing - similarity search
- ➔ • Data Mining

15-826 Copyright: C. Faloutsos (2006) 2

CMU SCS

Data Mining - Detailed outline

- Statistics
- ➔ • AI - decision trees
 - Problem
 - Approach
 - Conclusions
- DB

15-826 Copyright: C. Faloutsos (2006) 3

CMU SCS

Decision Trees

- Problem: Classification - I.e.,
- given a training set (N tuples, with M attributes, plus a label attribute)
- find rules, to predict the label for newcomers

Pictorially:

15-826 Copyright: C. Faloutsos (2006) 4

CMU SCS

Decision trees

Age	Chol-level	Gender	...	CLASS-ID
30	150	M		+
				...
				-
				??

15-826 Copyright: C. Faloutsos (2006) 5

CMU SCS

Decision trees

- issues:
 - missing values
 - noise
 - 'rare' events

15-826 Copyright: C. Faloutsos (2006) 6

CMU SCS

Decision trees

- types of attributes
 - numerical (= continuous) - eg: 'salary'
 - ordinal (= integer) - eg.: '# of children'
 - nominal (= categorical) - eg.: 'car-type'

15-826 Copyright: C. Faloutsos (2006) 7

CMU SCS

Decision trees

- Pictorially, we have

15-826 Copyright: C. Faloutsos (2006) 8

CMU SCS

Decision trees

- and we want to label '?'

15-826 Copyright: C. Faloutsos (2006) 9

CMU SCS

Decision trees

- so we build a decision tree:

15-826 Copyright: C. Faloutsos (2006) 10

CMU SCS

Decision trees

- so we build a decision tree:

15-826 Copyright: C. Faloutsos (2006) 11

CMU SCS

Data Mining - Detailed outline

- Statistics
- AI - decision trees
 - Problem
 - Approach
 - Conclusions
- DB

15-826 Copyright: C. Faloutsos (2006) 12

CMU SCS

Decision trees

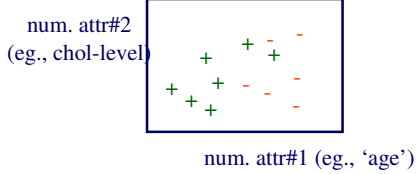
- Typically, two steps:
 - tree building
 - tree pruning (for over-training/over-fitting)

15-826 Copyright: C. Faloutsos (2006) 13

CMU SCS

Tree building

- How?



15-826 Copyright: C. Faloutsos (2006) 14

CMU SCS

Tree building

- How?
- A: Partition, recursively - pseudocode:


```
Partition ( Dataset S)
  if all points in S have same label
  then return
  evaluate splits along each attribute A
  pick best split, to divide S into S1 and S2
  Partition(S1); Partition(S2)
```

15-826 Copyright: C. Faloutsos (2006) 15

CMU SCS

Tree building

- Q1: how to introduce splits along attribute A_i
- Q2: how to evaluate a split?

15-826 Copyright: C. Faloutsos (2006) 16

CMU SCS

Tree building

- Q1: how to introduce splits along attribute A_i
- A1:
 - for num. attributes:
 - binary split, or
 - multiple split
 - for categorical attributes:
 - compute all subsets (expensive!), or
 - use a greedy algo

15-826 Copyright: C. Faloutsos (2006) 17

CMU SCS

Tree building

- Q1: how to introduce splits along attribute A_i
- ➔ Q2: how to evaluate a split?

15-826 Copyright: C. Faloutsos (2006) 18

Tree building

- Q1: how to introduce splits along attribute A_i
- ➔ • Q2: how to evaluate a split?
- A: by how close to uniform each subset is - ie., we need a measure of uniformity:

15-826 Copyright: C. Faloutsos (2006) 19

Tree building

entropy: $H(p_+, p_-)$ Any other measure?

15-826 Copyright: C. Faloutsos (2006) 20

Tree building

entropy: $H(p_+, p_-)$ 'gini' index: $1 - p_+^2 - p_-^2$

15-826 Copyright: C. Faloutsos (2006) 21

Tree building

entropy: $H(p_+, p_-)$ 'gini' index: $1 - p_+^2 - p_-^2$

(How about multiple labels?)

15-826 Copyright: C. Faloutsos (2006) 22

Tree building

Intuition:

- entropy: #bits to encode the class label
- gini: classification error, if we randomly guess '+' with prob. p_+

15-826 Copyright: C. Faloutsos (2006) 23

Tree building

Thus, we choose the split that reduces entropy/classification-error the most: Eg.:

15-826 Copyright: C. Faloutsos (2006) 24

CMU SCS

Tree building

- Before split: we need $(n_+ + n_-) * H(p_+, p_-) = (7+6) * H(7/13, 6/13)$ bits total, to encode all the class labels
- After the split we need:
 - 0 bits for the first half and $(2+6) * H(2/8, 6/8)$ bits for the second half

15-826 Copyright: C. Faloutsos (2006) 25

CMU SCS

Data Mining - Detailed outline

- Statistics
- AI - decision trees
 - Problem
 - Approach
 - tree building
 - tree pruning
 - Conclusions
- DB

15-826 Copyright: C. Faloutsos (2006) 26

CMU SCS

Tree pruning

- What for?

num. attr#2
(eg., chol-level)

num. attr#1 (eg., 'age')

15-826 Copyright: C. Faloutsos (2006) 27

CMU SCS

Tree pruning

- Q: How to do it?

num. attr#2
(eg., chol-level)

num. attr#1 (eg., 'age')

15-826 Copyright: C. Faloutsos (2006) 28

CMU SCS

Tree pruning

- Q: How to do it?
- A1: use a 'training' and a 'testing' set - prune nodes that improve classification in the 'testing' set. (Drawbacks?)

15-826 Copyright: C. Faloutsos (2006) 29

CMU SCS

Tree pruning

- Q: How to do it?
- A1: use a 'training' and a 'testing' set - prune nodes that improve classification in the 'testing' set. (Drawbacks?)
- A2: or, rely on MDL (= Minimum Description Language) - in detail:

15-826 Copyright: C. Faloutsos (2006) 30

CMU SCS

Tree pruning

- envision the problem as compression (of what?)

15-826 Copyright: C. Faloutsos (2006) 31

CMU SCS

Tree pruning

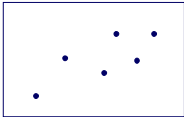
- envision the problem as compression (of what?)
- and try to min. the # bits to compress
 - the class labels AND
 - the representation of the decision tree

15-826 Copyright: C. Faloutsos (2006) 32

CMU SCS

(MDL)

- a brilliant idea - eg.: best n -degree polynomial to compress these points:
- the one that minimizes (sum of errors $\pm n$)



15-826 Copyright: C. Faloutsos (2006) 33

CMU SCS

Conclusions

- Classification through trees
- Building phase - splitting policies
- Pruning phase (to avoid over-fitting)
- Observation: classification is subtly related to compression

15-826 Copyright: C. Faloutsos (2006) 34

CMU SCS

Reference

- M. Mehta, R. Agrawal and J. Rissanen, '*SLIQ: A Fast Scalable Classifier for Data Mining*', Proc. of the Fifth Int'l Conference on Extending Database Technology, Avignon, France, March 1996

15-826 Copyright: C. Faloutsos (2006) 35