

CMU SCS

15-826: Multimedia Databases and Data Mining

Lecture #29: Data Mining - trees + assoc. rules
C. Faloutsos

CMU SCS

Must-read Material

- Agrawal, R., S. Ghosh, et al. (Aug. 23-27, 1992). An Interval Classifier for Database Mining Applications. VLDB Conf. Proc., Vancouver, BC, Canada.
- Han + Kamber, chapter 6.1-4 (1st Edition); or Chapter 5.1-4 (2nd Edition)
- Mehta, M., R. Agrawal, et al. (1996). SLIQ: A Fast Scalable Classifier for Data Mining. EDBT, Avignon, France.

15-826 Copyright: C. Faloutsos (2010) 2

CMU SCS

Must-read Material

- Rakesh Agrawal, Tomasz Imielinski and Arun Swami *Mining Association Rules Between Sets of Items in Large Databases* Proc. ACM SIGMOD, Washington, DC, May 1993, pp. 207-216

15-826 Copyright: C. Faloutsos (2010) 3

CMU SCS

Outline

Goal: 'Find similar / interesting things'

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

15-826 Copyright: C. Faloutsos (2010) 4

CMU SCS

Data Mining - Detailed outline

- data warehouses; data cubes; OLAP
- ➔ classifiers
- association rules

15-826 Copyright: C. Faloutsos (2010) 5

CMU SCS

**NOT in
Final Exam**

Classifiers - outline

- ➔ Case study: 'Interval Classifier' ('IC')
- recent developments and variations

15-826 Copyright: C. Faloutsos (2010) 6

CMU SCS **NOT in Final Exam**

Tree Classifiers

Database issues: how about huge (training) datasets?

Case study: Interval Classifier [Agrawal+92]
 Goal: build a classifier (eg., for target mailing)
 Differences from AI/ML:

- retrieval efficiency (could use DBMS indices!)
- generation efficiency (large training dataset)

15-826 Copyright: C. Faloutsos (2010) 7

CMU SCS **NOT in Final Exam**

Tree Classifiers - 'IC'

Proposed method: use classification tree, but

- split a range (= num. attribute) into k sub-ranges, as opposed to just 2
- do 'dynamic pruning' (ie., don't expand a node that is fairly homogeneous)

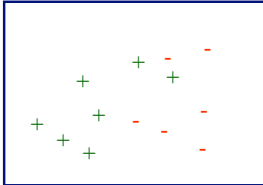
15-826 Copyright: C. Faloutsos (2010) 8

CMU SCS **NOT in Final Exam**

Decision trees

- Pictorially, we have

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



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

15-826 Copyright: C. Faloutsos (2010) 9

CMU SCS **NOT in Final Exam**

Decision trees

- and we want to label '?'

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

?

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

15-826 Copyright: C. Faloutsos (2010) 10

CMU SCS **NOT in Final Exam**

Decision trees

- so we build a decision tree:

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

40

?

50
num. attr#1 (eg., 'age')

15-826 Copyright: C. Faloutsos (2010) 11

CMU SCS **NOT in Final Exam**

Tree Classifiers - 'IC'

Sketch of algorithm

make-tree():

- partition set in groups by label
- obtain histograms for each group and each attribute
- Apply goodness function to pick winning attribute A'
- Partition the domain of A' into "strong" and "weak" intervals
- For each "strong" interval: assign it to majority label
- For each "weak" interval: make-tree()

15-826 Copyright: C. Faloutsos (2010) 12

CMU SCS **NOT in Final Exam**

Tree Classifiers - 'IC'

- "strong" interval: = homogeneous (or close enough)
- k : depends on # of distinct values
- 'interval' = 'range' for a continuous attribute;
- 'interval' = 'value' for a categorical one
- histograms: equi-width

Classification accuracy: comparable to standard algorithms (ID3, C4)

15-826 Copyright: C. Faloutsos (2010) 13

CMU SCS **NOT in Final Exam**

Tree Classifiers - 'IC'

Conclusions: compared to standard algorithms (ID3, C4):

- Faster, because of
 - k -way splitting and
 - dynamic pruning
- comparable classification accuracy

15-826 Copyright: C. Faloutsos (2010) 14

CMU SCS **NOT in Final Exam**

Classifiers - outline

- Case study: 'Interval Classifier' ('IC')
- ➔ variations

15-826 Copyright: C. Faloutsos (2010) 15

CMU SCS **NOT in Final Exam**

Classifiers - newer methods

- SLIQ [Mehta+96]
- SPRINT [Shafer+, vldb96]
- PUBLIC [Rastogi+Shim, vldb98]
- RainForest [Gehrke+, 2000]

Goal: how to make build decision trees, when the training set does not fit in memory

15-826 Copyright: C. Faloutsos (2010) 16

CMU SCS **NOT in Final Exam**

Classifiers - newer methods

Goal: how to make build decision trees, when the training set does not fit in memory

SLIQ: use vertical partitioning (att-value, record-id) for each attribute; keep the (label, record-id) list in main memory

SPRINT: like SLIQ, but attach 'label' on each attribute list: (attr-value, label, record-id)

15-826 Copyright: C. Faloutsos (2010) 17

CMU SCS **NOT in Final Exam**

Classifiers - conclusions

Variations: try to improve scalability/speed with

- 'dynamic' pruning
- elaborate file structures / data placement
- parallelism

15-826 Copyright: C. Faloutsos (2010) 18

CMU SCS

Data Mining - Detailed outline

- data warehouses; data cubes; OLAP
- classifiers
- ➔ association rules

15-826 Copyright: C. Faloutsos (2010) 19

CMU SCS

Association rules - outline

- ➔ Main idea [Agrawal+SIGMOD93]
 - performance improvements
 - Variations / Applications
 - Follow-up concepts

15-826 Copyright: C. Faloutsos (2010) 20

CMU SCS

Association rules - idea

[Agrawal+SIGMOD93]

- Consider 'market basket' case:
 - (milk, bread)
 - (milk)
 - (milk, chocolate)
 - (milk, bread)
- Find 'interesting things', eg., rules of the form:
 - milk, bread -> chocolate | 90%

15-826 Copyright: C. Faloutsos (2010) 21

CMU SCS

Association rules - idea

In general, for a given rule
 $I_j, I_k, \dots, I_m \rightarrow I_x | c$
 'c' = 'confidence' (how often people buy I_x , given that they have bought I_j, \dots, I_m)
 's' = support: how often people buy I_j, \dots, I_m, I_x

15-826 Copyright: C. Faloutsos (2010) 22

CMU SCS

Association rules - idea

Problem definition:

- given
 - a set of 'market baskets' (=binary matrix, of N rows /baskets and M columns/products)
 - min-support 's' and
 - min-confidence 'c'
- find
 - all the rules with higher support and confidence

15-826 Copyright: C. Faloutsos (2010) 23

CMU SCS

Association rules - idea

Closely related concept: "large itemset"
 $I_j, I_k, \dots, I_m, I_x$
 is a 'large itemset', if it appears more than 'min-support' times

Observation: once we have a 'large itemset', we can find out the qualifying rules easily (how?)
 Thus, let's focus on how to find 'large itemsets'

15-826 Copyright: C. Faloutsos (2010) 24

CMU SCS

Association rules - idea

Naive solution: scan database once; keep $2^{*|I|}$ counters
 Drawback?
 Improvement?

15-826 Copyright: C. Faloutsos (2010) 25

CMU SCS

Association rules - idea

Naive solution: scan database once; keep $2^{*|I|}$ counters
 Drawback? 2^{*1000} is prohibitive...
 Improvement? scan the db $|I|$ times, looking for 1-, 2-, etc itemsets

Eg., for $|I|=3$ items only (A, B, C), we have

15-826 Copyright: C. Faloutsos (2010) 26

CMU SCS

Association rules - idea

(A)

100

(B)

200

(C)

2

first pass

min-sup:10

15-826 Copyright: C. Faloutsos (2010) 27

CMU SCS

Association rules - idea

first pass
min-sup:10

15-826 Copyright: C. Faloutsos (2010) 28

CMU SCS

Association rules - idea

Anti-monotonicity property:
if an itemset fails to be 'large', so will every superset of it (hence all supersets can be pruned)

Sketch of the (famous!) 'a-priori' algorithm
Let $L(i-1)$ be the set of large itemsets with $i-1$ elements
Let $C(i)$ be the set of candidate itemsets (of size i)

15-826 Copyright: C. Faloutsos (2010) 29

CMU SCS

Association rules - idea

Compute $L(1)$, by scanning the database.
repeat, for $i=2,3,\dots$,
 'join' $L(i-1)$ with itself, to generate $C(i)$
 two itemset can be joined, if they agree on their first $i-2$ elements
 prune the itemsets of $C(i)$ (how?)
 scan the db, finding the counts of the $C(i)$ itemsets - set this to be $L(i)$
 unless $L(i)$ is empty, repeat the loop
(see example 6.1 in [Han+Kamber])

15-826 Copyright: C. Faloutsos (2010) 30

CMU SCS

Association rules - outline

- Main idea [Agrawal+SIGMOD93]
- ▶ performance improvements
- Variations / Applications
- Follow-up concepts

15-826 Copyright: C. Faloutsos (2010) 31

CMU SCS

Association rules - improvements

- Use the independence assumption, to second-guess large itemsets a few steps ahead
- eliminate 'market baskets', that don't contain any more large itemsets
- Partitioning (eg., for parallelism): find 'local large itemsets', and merge.
- Sampling
- report only 'maximal large itemsets' (dfn?)
- FP-tree (seems to be the fastest)

15-826 Copyright: C. Faloutsos (2010) 32

CMU SCS

Association rules - improvements

- FP-tree: no candidate itemset generation - only two passes over dataset
- Main idea: build a TRIE in main memory

Specifically:

- first pass, to find counts of each item - sort items in decreasing count order
- second pass: build the TRIE, and update its counts

(eg., let A,B, C, D be the items in frequency order:)

15-826 Copyright: C. Faloutsos (2010) 33

CMU SCS

Association rules - improvements

- eg., let A,B, C, D be the items in frequency order:)

```

graph TD
    Root((32)) --- A((10))
    Root --- Empty1(( ))
    Root --- Empty2(( ))
    Root --- Empty3(( ))
    Root --- Empty4(( ))
    Root --- Empty5(( ))
    Root --- Empty6(( ))
    Root --- Empty7(( ))
    Root --- Empty8(( ))
    Root --- Empty9(( ))
    Root --- Empty10(( ))
    Root --- Empty11(( ))
    Root --- Empty12(( ))
    Root --- Empty13(( ))
    Root --- Empty14(( ))
    Root --- Empty15(( ))
    Root --- Empty16(( ))
    Root --- Empty17(( ))
    Root --- Empty18(( ))
    Root --- Empty19(( ))
    Root --- Empty20(( ))
    Root --- Empty21(( ))
    Root --- Empty22(( ))
    Root --- Empty23(( ))
    Root --- Empty24(( ))
    Root --- Empty25(( ))
    Root --- Empty26(( ))
    Root --- Empty27(( ))
    Root --- Empty28(( ))
    Root --- Empty29(( ))
    Root --- Empty30(( ))
    Root --- Empty31(( ))
    Root --- Empty32(( ))
    A --- B((4))
    A --- AC((2))
    A --- Empty33(( ))
    A --- Empty34(( ))
    A --- Empty35(( ))
    A --- Empty36(( ))
    A --- Empty37(( ))
    A --- Empty38(( ))
    A --- Empty39(( ))
    A --- Empty40(( ))
    A --- Empty41(( ))
    A --- Empty42(( ))
    A --- Empty43(( ))
    A --- Empty44(( ))
    A --- Empty45(( ))
    A --- Empty46(( ))
    A --- Empty47(( ))
    A --- Empty48(( ))
    A --- Empty49(( ))
    A --- Empty50(( ))
    A --- Empty51(( ))
    A --- Empty52(( ))
    A --- Empty53(( ))
    A --- Empty54(( ))
    A --- Empty55(( ))
    A --- Empty56(( ))
    A --- Empty57(( ))
    A --- Empty58(( ))
    A --- Empty59(( ))
    A --- Empty60(( ))
    A --- Empty61(( ))
    A --- Empty62(( ))
    A --- Empty63(( ))
    A --- Empty64(( ))
    A --- Empty65(( ))
    A --- Empty66(( ))
    A --- Empty67(( ))
    A --- Empty68(( ))
    A --- Empty69(( ))
    A --- Empty70(( ))
    A --- Empty71(( ))
    A --- Empty72(( ))
    A --- Empty73(( ))
    A --- Empty74(( ))
    A --- Empty75(( ))
    A --- Empty76(( ))
    A --- Empty77(( ))
    A --- Empty78(( ))
    A --- Empty79(( ))
    A --- Empty80(( ))
    A --- Empty81(( ))
    A --- Empty82(( ))
    A --- Empty83(( ))
    A --- Empty84(( ))
    A --- Empty85(( ))
    A --- Empty86(( ))
    A --- Empty87(( ))
    A --- Empty88(( ))
    A --- Empty89(( ))
    A --- Empty90(( ))
    A --- Empty91(( ))
    A --- Empty92(( ))
    A --- Empty93(( ))
    A --- Empty94(( ))
    A --- Empty95(( ))
    A --- Empty96(( ))
    A --- Empty97(( ))
    A --- Empty98(( ))
    A --- Empty99(( ))
    A --- Empty100(( ))
    AC --- C((1))
    AC --- Empty101(( ))
    AC --- Empty102(( ))
    AC --- Empty103(( ))
    AC --- Empty104(( ))
    AC --- Empty105(( ))
    AC --- Empty106(( ))
    AC --- Empty107(( ))
    AC --- Empty108(( ))
    AC --- Empty109(( ))
    AC --- Empty110(( ))
    AC --- Empty111(( ))
    AC --- Empty112(( ))
    AC --- Empty113(( ))
    AC --- Empty114(( ))
    AC --- Empty115(( ))
    AC --- Empty116(( ))
    AC --- Empty117(( ))
    AC --- Empty118(( ))
    AC --- Empty119(( ))
    AC --- Empty120(( ))
    AC --- Empty121(( ))
    AC --- Empty122(( ))
    AC --- Empty123(( ))
    AC --- Empty124(( ))
    AC --- Empty125(( ))
    AC --- Empty126(( ))
    AC --- Empty127(( ))
    AC --- Empty128(( ))
    AC --- Empty129(( ))
    AC --- Empty130(( ))
    AC --- Empty131(( ))
    AC --- Empty132(( ))
    AC --- Empty133(( ))
    AC --- Empty134(( ))
    AC --- Empty135(( ))
    AC --- Empty136(( ))
    AC --- Empty137(( ))
    AC --- Empty138(( ))
    AC --- Empty139(( ))
    AC --- Empty140(( ))
    AC --- Empty141(( ))
    AC --- Empty142(( ))
    AC --- Empty143(( ))
    AC --- Empty144(( ))
    AC --- Empty145(( ))
    AC --- Empty146(( ))
    AC --- Empty147(( ))
    AC --- Empty148(( ))
    AC --- Empty149(( ))
    AC --- Empty150(( ))
    AC --- Empty151(( ))
    AC --- Empty152(( ))
    AC --- Empty153(( ))
    AC --- Empty154(( ))
    AC --- Empty155(( ))
    AC --- Empty156(( ))
    AC --- Empty157(( ))
    AC --- Empty158(( ))
    AC --- Empty159(( ))
    AC --- Empty160(( ))
    AC --- Empty161(( ))
    AC --- Empty162(( ))
    AC --- Empty163(( ))
    AC --- Empty164(( ))
    AC --- Empty165(( ))
    AC --- Empty166(( ))
    AC --- Empty167(( ))
    AC --- Empty168(( ))
    AC --- Empty169(( ))
    AC --- Empty170(( ))
    AC --- Empty171(( ))
    AC --- Empty172(( ))
    AC --- Empty173(( ))
    AC --- Empty174(( ))
    AC --- Empty175(( ))
    AC --- Empty176(( ))
    AC --- Empty177(( ))
    AC --- Empty178(( ))
    AC --- Empty179(( ))
    AC --- Empty180(( ))
    AC --- Empty181(( ))
    AC --- Empty182(( ))
    AC --- Empty183(( ))
    AC --- Empty184(( ))
    AC --- Empty185(( ))
    AC --- Empty186(( ))
    AC --- Empty187(( ))
    AC --- Empty188(( ))
    AC --- Empty189(( ))
    AC --- Empty190(( ))
    AC --- Empty191(( ))
    AC --- Empty192(( ))
    AC --- Empty193(( ))
    AC --- Empty194(( ))
    AC --- Empty195(( ))
    AC --- Empty196(( ))
    AC --- Empty197(( ))
    AC --- Empty198(( ))
    AC --- Empty199(( ))
    AC --- Empty200(( ))
    C --- Empty201(( ))
    C --- Empty202(( ))
    C --- Empty203(( ))
    C --- Empty204(( ))
    C --- Empty205(( ))
    C --- Empty206(( ))
    C --- Empty207(( ))
    C --- Empty208(( ))
    C --- Empty209(( ))
    C --- Empty210(( ))
    C --- Empty211(( ))
    C --- Empty212(( ))
    C --- Empty213(( ))
    C --- Empty214(( ))
    C --- Empty215(( ))
    C --- Empty216(( ))
    C --- Empty217(( ))
    C --- Empty218(( ))
    C --- Empty219(( ))
    C --- Empty220(( ))
    C --- Empty221(( ))
    C --- Empty222(( ))
    C --- Empty223(( ))
    C --- Empty224(( ))
    C --- Empty225(( ))
    C --- Empty226(( ))
    C --- Empty227(( ))
    C --- Empty228(( ))
    C --- Empty229(( ))
    C --- Empty230(( ))
    C --- Empty231(( ))
    C --- Empty232(( ))
    C --- Empty233(( ))
    C --- Empty234(( ))
    C --- Empty235(( ))
    C --- Empty236(( ))
    C --- Empty237(( ))
    C --- Empty238(( ))
    C --- Empty239(( ))
    C --- Empty240(( ))
    C --- Empty241(( ))
    C --- Empty242(( ))
    C --- Empty243(( ))
    C --- Empty244(( ))
    C --- Empty245(( ))
    C --- Empty246(( ))
    C --- Empty247(( ))
    C --- Empty248(( ))
    C --- Empty249(( ))
    C --- Empty250(( ))
    C --- Empty251(( ))
    C --- Empty252(( ))
    C --- Empty253(( ))
    C --- Empty254(( ))
    C --- Empty255(( ))
    C --- Empty256(( ))
    C --- Empty257(( ))
    C --- Empty258(( ))
    C --- Empty259(( ))
    C --- Empty260(( ))
    C --- Empty261(( ))
    C --- Empty262(( ))
    C --- Empty263(( ))
    C --- Empty264(( ))
    C --- Empty265(( ))
    C --- Empty266(( ))
    C --- Empty267(( ))
    C --- Empty268(( ))
    C --- Empty269(( ))
    C --- Empty270(( ))
    C --- Empty271(( ))
    C --- Empty272(( ))
    C --- Empty273(( ))
    C --- Empty274(( ))
    C --- Empty275(( ))
    C --- Empty276(( ))
    C --- Empty277(( ))
    C --- Empty278(( ))
    C --- Empty279(( ))
    C --- Empty280(( ))
    C --- Empty281(( ))
    C --- Empty282(( ))
    C --- Empty283(( ))
    C --- Empty284(( ))
    C --- Empty285(( ))
    C --- Empty286(( ))
    C --- Empty287(( ))
    C --- Empty288(( ))
    C --- Empty289(( ))
    C --- Empty290(( ))
    C --- Empty291(( ))
    C --- Empty292(( ))
    C --- Empty293(( ))
    C --- Empty294(( ))
    C --- Empty295(( ))
    C --- Empty296(( ))
    C --- Empty297(( ))
    C --- Empty298(( ))
    C --- Empty299(( ))
    C --- Empty300(( ))
  
```

32 records
10 of them have A
4 have AB
2 have AC
1 has C

15-826 Copyright: C. Faloutsos (2010) 34

CMU SCS

Association rules - improvements

- Traversing the TRIE, we can find the large itemsets (details: in [Han+Kamber, §6.2.4])
- Result: much faster than 'a-priori' (order of magnitude)

15-826 Copyright: C. Faloutsos (2010) 35

CMU SCS

Association rules - outline

- Main idea [Agrawal+SIGMOD93]
- performance improvements
- Variations / Applications
- Follow-up concepts

15-826 Copyright: C. Faloutsos (2010) 36

CMU SCS

Association rules - variations

1) Multi-level rules: given concept hierarchy

- 'bread', 'milk', 'butter' -> foods;
- 'aspirin', 'tylenol' -> pharmacy

look for rules across any level of the hierarchy, eg
 'aspirin' -> foods

(similarly, rules across dimensions, like 'product',
 'time', 'branch':
 'bread', '12noon', 'PGH-branch' -> 'milk')

15-826 Copyright: C. Faloutsos (2010) 37

CMU SCS

Association rules - variations

2) Sequential patterns:

'car', 'now' -> 'tires', '2 months later'

Also: given a stream of (time-stamped) events:
 A A B A C A B A C

find rules like
 B, A -> C

[Manilla+KDD97]

15-826 Copyright: C. Faloutsos (2010) 38

CMU SCS

Association rules - variations

3) Spatial rules, eg:
 'house close to lake' -> 'expensive'

15-826 Copyright: C. Faloutsos (2010) 39

CMU SCS

Association rules - variations

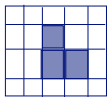
4) Quantitative rules, eg:
 'age between 20 and 30', 'chol. level <150' ->
 'weight > 150lb'
 Ie., given **numerical** attributes, how to find rules?

15-826 Copyright: C. Faloutsos (2010) 40

CMU SCS

Association rules - variations

4) Quantitative rules
 Solution:
 • bucketize the (numerical) attributes
 • find (binary) rules
 • stitch appropriate buckets together:

salary  age

15-826 Copyright: C. Faloutsos (2010) 41

CMU SCS

Association rules - outline

- Main idea [Agrawal+SIGMOD93]
- performance improvements
- Variations / Applications
- ➔ Follow-up concepts

15-826 Copyright: C. Faloutsos (2010) 42

CMU SCS

Association rules - follow-up concepts

Associations rules vs. correlation.
 Motivation: if
 milk, bread
 is a 'large itemset', does this means that there is a positive correlation between 'milk' and 'bread' sales?

15-826 Copyright: C. Faloutsos (2010) 43

CMU SCS

Association rules - follow-up concepts

What to do, then?


15-826 Copyright: C. Faloutsos (2010) 44

CMU SCS

Association rules - follow-up concepts

What to do, then?
 A: report only pairs of items that are indeed correlated - ie, they pass the Chi-square test
 The idea can be extended to 3-, 4- etc itemsets (but becomes more expensive to check)
 See [Han+Kamber, §6.5], or [Brin+,SIGMOD97]

15-826 Copyright: C. Faloutsos (2010) 45

 CMU SCS

Association rules - Conclusions

Association rules: a new tool to find patterns

- easy to understand its output
- fine-tuned algorithms exist

15-826 Copyright: C. Faloutsos (2010) 46
