Fast Counting

The Auton Lab

Carnegie Mellon University

www.autonlab.org

Counting and other Additive Aggregates

- Suppose you have a dataset with 40 attributes and a million records.
- Suppose the attributes are all categorical (discrete)
- You keep asking (billions of) questions like
  - *"what's the mutual information between height, agegroup, education"*
  - *"Is there a 5-d correlation between cell-morphology, nucleus-change, attack site, cell-type and exposure time?"*
  - *"does the average wealth of inner-city blue-collar SUV owners seem to be correlated with their health?"*

Example applications

- Bayesian Networks
- Causal Networks
- Rule Learning
- Feature Selecting
- GMDH
- Frequent Sets
- Decision Tree Learning
- Decision List Learning
- TFIDF
- Bayes Classifiers
- Logistic Regression

A Categorical (Binary) Dataset

Wealthy	Healthy	Happy
1	0	1
0	1	0
0	1	1
0	0	1
1	1	1
0	1	1
1	1	1

R=7 Records

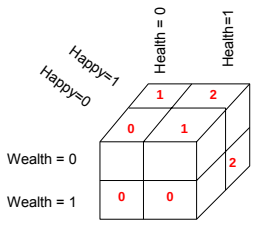
M=3 Attributes

Datacubes

Wealthy	Healthy	Happy
1	0	1
0	1	0
0	1	1
0	0	1
1	1	1
0	1	1
1	1	1

R=7 Records

M=3 Attributes



Datacubes

Wealthy	Healthy	Happ
1	0	1
0	1	0
0	1	1
0	0	1
1	1	1
0	1	1
1	1	1

R=7 Records

M=3 Attributes

Linda Poggit, Data Miner

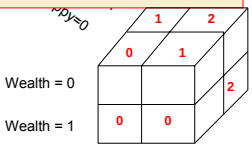
"I call that a Datacube"

Jerry Gold, Statistician

"I call that a Contingency Table"

Lenny Welk, Computer Scientist

"I call that a multidimensional array of counts"



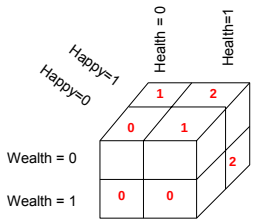
Datacubes

Wealthy	Healthy	Happy
1	0	1
0	1	0
0	1	1
0	0	1
1	1	1
0	1	1
1	1	1

R=7 Records

M=3 Attributes

A Datacube is a simple effective cached representation of categorical data!



7

Datacubes

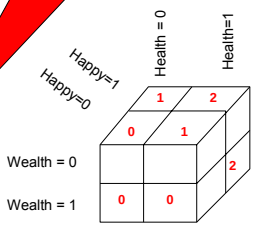
Wealthy	Healthy	Happy
1	0	1
0	1	0
0	1	1
0	0	1
1	1	1
0	1	1
1	1	1

R=7 Records

M=3 Attributes

Problem: Things get uncomfortable with (say) 10^6 records and 100 attributes…

A Datacube is a simple effective cached representation of categorical data!



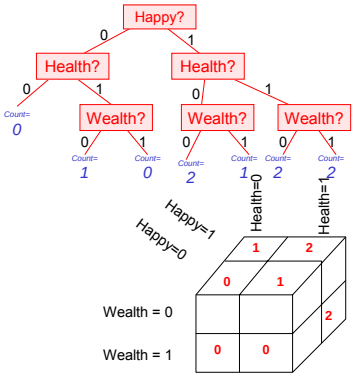
8

Sparse Representation of Datacubes

Wealthy	Healthy	Happy
1	0	1
0	1	0
0	1	1
0	0	1
1	1	1
0	1	1
1	1	1

R=7 Records

M=3 Attributes



9

Sparse Representation of Datacubes

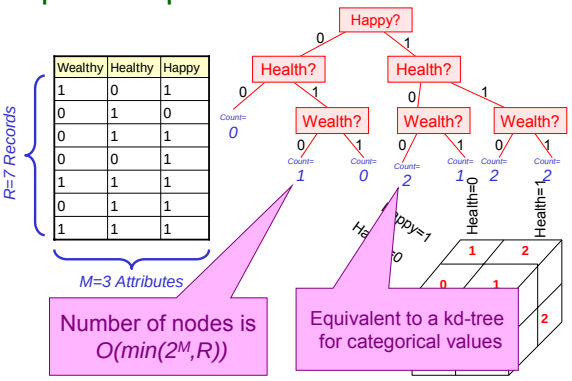
Wealthy	Healthy	Happy
1	0	1
0	1	0
0	1	1
0	0	1
1	1	1
0	1	1
1	1	1

R=7 Records

M=3 Attributes

Number of nodes is $O(min(2^M,R))$

Equivalent to a kd-tree for categorical values



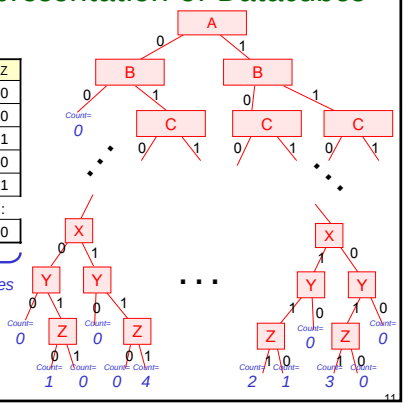
10

Sparse Representation of Datacubes

A	B	C	...	Y	Z
1	0	1	...	1	0
0	1	0	...	1	0
0	1	0	...	1	1
0	0	1	...	0	0
1	1	0	...	1	1
:	:	:	:	:	:
1	1	0	...	1	0

R=100,000 Records

M=26 Attributes



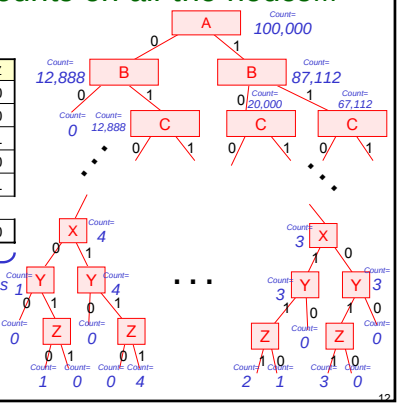
11

Putting Counts on all the nodes...

A	B	C	...	Y	Z
1	0	1	...	1	0
0	1	0	...	1	0
0	1	0	...	1	1
0	0	1	...	0	0
1	1	0	...	1	1
:	:	:	:	:	:
1	1	0	...	1	0

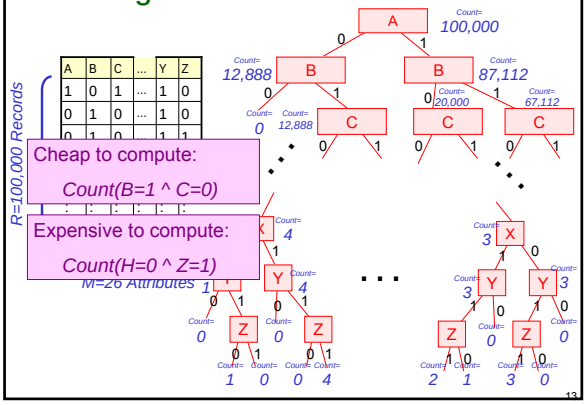
R=100,000 Records

M=26 Attributes



12

Putting Counts on all the nodes...



13

Sparse Datasets

	Anna	Brigham	Dan	Jean	Paul	Remi	Sajid	Ting
Link1	1	1	0	0	1	0	0	1
Link2	1	1	0	0	0	0	0	0
Link3	0	1	1	0	0	0	1	0
Link4	0	0	1	0	0	0	1	0
Link5	1	0	0	0	0	0	0	0
Link6	1	0	0	0	0	0	0	1
Link7	0	0	1	1	1	0	0	0
Link8	0	1	0	0	1	0	1	1
:	:	:	:	:	:	:	:	:

14

Sparse Datasets

	Anna	Brigham	Dan	Jean	Paul	Remi	Sajid	Ting
Link1	1	1	0	0	1	0	0	1
Link2	1	1	0	0	0	0	0	0
Link3	0	1	1	0	0	0	1	0
Link4	0	0	1	0	0	0	1	0
Link5	1	0	0	0	0	0	0	0
Link6	1	0	0	0	0	0	0	1
Link7	0	0	1	1	1	0	0	0
Link8	0	1	0	0	1	0	1	1
:	:	:	:	:	:	:	:	:

Which single attributes occur?	A(4) B(4) D(3) J(1) P(3) S(3) T(3)
Which pairs co-occur?	AB(2) AP(1) AT(2) BD(1) BP(2) BS(2) BT(2) DJ(1) DP(1) DS(2) JP(1) PS(1) PT(2) ST(1)
Which triples co-occur?	ABP(1) ABT(1) BDS(1) BPS(1) BPT(2) DJP(1) BST(1) PST(1)
Quadruples?	ABPT(1) BPST(1)
Quintuples?	none

15

This allows us to answer

Count(X_1=1 ^ X_2=1 ^ .. X_k = 1)

in constant time for any subset of attributes

Which single attributes occur?	A(4) B(4) D(3) J(1) P(3) S(3) T(3)
Which pairs co-occur?	AB(2) AP(1) AT(2) BD(1) BP(2) BS(2) BT(2) DJ(1) DP(1) DS(2) JP(1) PS(1) PT(2) ST(1)
Which triples co-occur?	ABP(1) ABT(1) BDS(1) BPS(1) BPT(2) DJP(1) BST(1) PST(1)
Quadruples?	ABPT(1) BPST(1)
Quintuples?	none

16

This allows us to answer

Count

in constant time for any subset of attributes

In fact, we can answer

Count(X_1=v_1 ^ X_2=v_2 ^ .. X_k = v_k)

in constant time for any subset of attributes and any assignment of 1's and 0's to v_1, v_2 ... v_k

Which single attributes occur?	A(4) B(4) D(3) J(1) P(3) S(3) T(3)
Which pairs co-occur?	AB(2) AP(1) AT(2) BD(1) BP(2) BS(2) BT(2) DJ(1) DP(1) DS(2) JP(1) PS(1) PT(2) ST(1)
Which triples co-occur?	ABP(1) ABT(1) BDS(1) BPS(1) BPT(2) DJP(1) BST(1) PST(1)
Quadruples?	ABPT(1) BPST(1)
Quintuples?	none

17

This allows us to answer

Count

in constant time for any subset of attributes

In fact, we can answer

Count(X_1=v_1 ^ X_2=v_2 ^ .. X_k = v_k)

in constant time for any subset of attributes and any assignment of 1's and 0's to v_1, v_2 ... v_k

C(B=0 ^ P=1 ^ S=0) =

C(P=1 ^ S=0) – C(B=1 ^ P=1 ^ S=0) =

[C(P=1) – C(P=1 ^ S=1)] –

[C(B=1 ^ P=1) - C(B=1 ^ P=1 ^ S=1)]

Generalization of this example gives proof that all information about dataset counts have been saved in the co-occuring sets.

Which single attributes occ	
Which pairs c occur?	2)
Which triples occur?	(1)
Quadruples?	
Quintuples?	

18

Frequent Sets

	Anna	Brigham	Dan	Jean	Paul	Remi	Sajid	Ting
Link1	1	1	0	0	1	0	0	1
Link2	1	1	0	0	0	0	0	0
Link3	0	1	1	0	0	0	1	0
Link4	0	0	1	0	0	0	1	0
Link5	1	0	0	0	0	0	0	0
Link6	1	0	0	0	0	0	0	1
Link7	0	0	1	1	1	0	0	0
Link8	0	1	0	0	1	0	1	1
:	:	:	:	:	:	:	:	:

Which single attributes have support=2?	A(4) B(4) D(3) P(3) S(3) T(3)
Which pairs have support=2?	AB(2) AT(2) BP(2) BS(2) BT(2) DS(2) PT(2)
Which triples have support=2?	BPT(2)
Quadruples?	none
Quintuples?	none

A set of attributes is a *frequent set with support k* if they co-occur *k* or more times.

19

Frequent Sets

	Anna	Brigham	Dan	Jean	Paul	Remi	Sajid	Ting
Link1	1	1	0	0	1	0	0	1
Link2	1	1	0	0	0	0	0	0
Link3	0	1	1	0	0	0	1	0
Link4	0	0	1	0	0	0	1	0
Link5	1	0	0	0	0	0	0	0
Link6	1	0	0	0	0	0	0	1
Link7	0	0	1	1	1	0	0	0
Link8	0	1	0	0	1	0	1	1
:	:	:	:	:	:	:	:	:

Which single attributes have support=2?	A(4) B(4) D(3) P(3) S(3) T(3)
Which pairs have support=2?	AB(2) AT(2) BP(2) BS(2) BT(2) DS(2) PT(2)
Which triples have support=2?	BPT(2)
Quadruples?	none
Quintuples?	none

Can be computed quickly using (e.g.) Apriori algorithm (Agrawal et al, 1993)

A set of attributes is a *frequent set with support k* if they co-occur *k* or more times.

20

	Exact	Approximate is okay
Dense, thin	Dense Datacube	
Dense, fat	Tree or Hash Datacube	
Sparse	Co-occurring tuples	Frequent Sets

21

	Exact	Approximate is okay
Dense, thin	Dense Datacube	
Dense, fat	Tree or Hash Datacube	
Sparse	Co-occurring tuples	Frequent Sets

But often much too much memory

But often much too slow

But often much too much memory

- Manilla and Toivonen, 1996
- Hollmen et al, 2003
- Goldenberg ICML 2004

Exact for association rules

22

	Exact	Approximate is okay
Dense, thin	Dense Datacube	
Dense, fat	Tree or Hash Datacube	
Sparse	Co-occurring tuples	Frequent Sets

Or AD-Tree

23

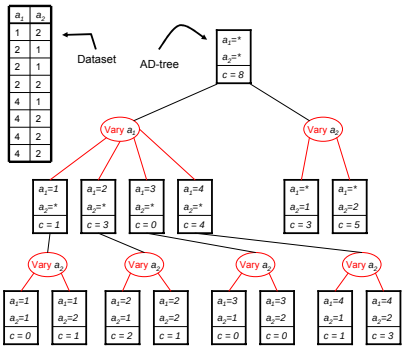
A crazy idea

Why not precompute the answers to all possible questions in advance?

Then, when new questions come in we can immediately look up the answer.

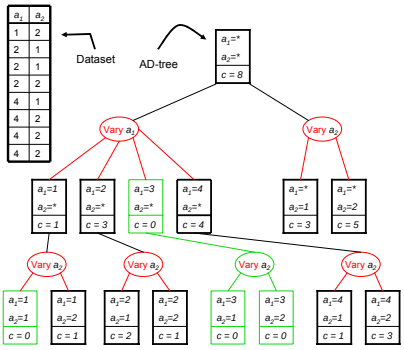
24

AD-Tree v1.0 Example



25

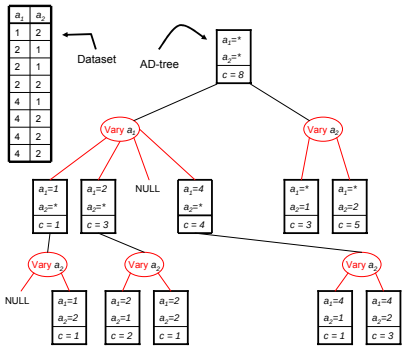
AD-Tree v1.1



A blindingly obvious improvement is to save memory by storing NULL pointers for counts of zero.

26

AD-Tree v1.1



A blindingly obvious improvement is to save memory by storing NULL pointers for counts of zero.

27

This is a crazy idea

With the census dataset, would need 400 Terabytes of memory

With the Birth dataset, would need…

28

This is a crazy idea

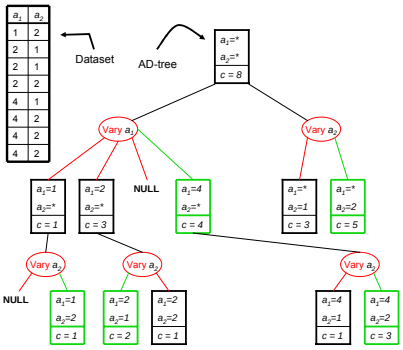
With the census dataset, would need 400 Terabytes of memory

With the Birth dataset, would need…

10^{23} terabytes of memory

29

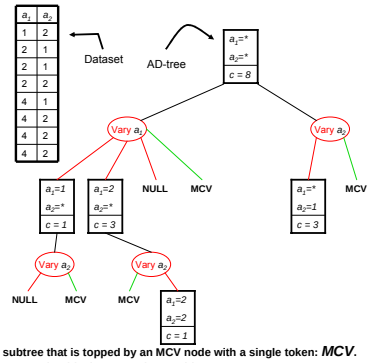
AD-Tree Version 2.0



For every Vary node in the tree, consider the Most Common Value.

30

AD-Tree Version 2.0

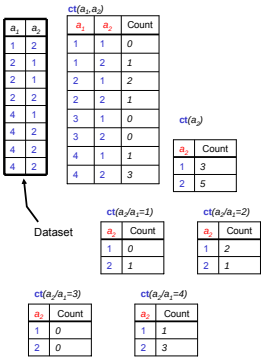


Replace any subtree that is topped by an MCV node with a single token: *MCV*. Wonderfully, counterintuitively, you've lost nothing!!!!

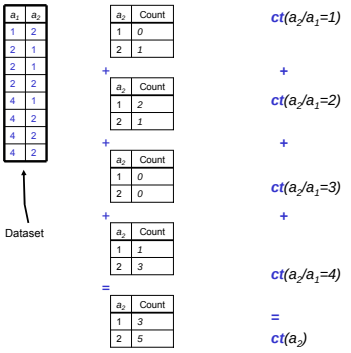
Notation Reminder

$ct(a_{i(1)}, a_{i(2)}, \ldots a_{i(n)})$ is the *contingency table* for attribute-set $\{a_{i(1)}, a_{i(2)}, \ldots a_{i(n)}\}$.

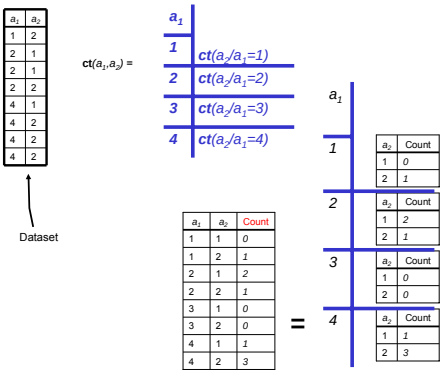
$ct(a_{i(1)}, a_{i(2)}, \ldots a_{i(n)} | a_{j(1)}=v_1, a_{j(2)}=v_2, \ldots a_{j(p)}=v_p)$ is a *conditional contingency table*: It's the contingency table for $a_{i(1)}, a_{i(2)}, \ldots a_{i(n)}$ restricted to those records that match $a_{j(1)}=v_1, a_{j(2)}=v_2, \ldots a_{j(p)}=v$



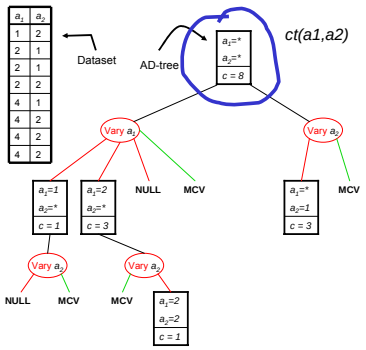
Contingency Algebra



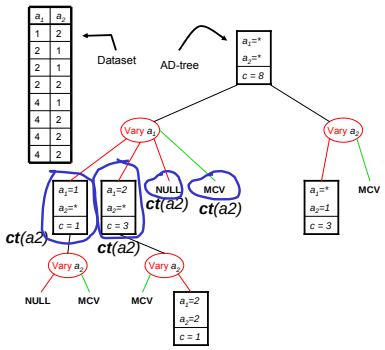
Contingency Algebra



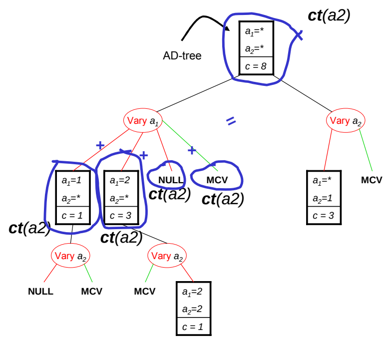
Computing $ct(a_1,a_2)$



Computing $ct(a_1,a_2)$



Computing ct(a_1,a_2)



37

Memory Use

Dataset	#bytes without the "MCV" trick	#bytes with the "MCV" trick
Census	10^{14}	10^{7}
Birth	10^{35}	10^{6}
m binary attributes: best case	2^{m}	m
m binary attributes: worst case	3^{m}	(R = # records) $\sum_{k=1}^{\log_2(R)} \binom{m}{k}$

38

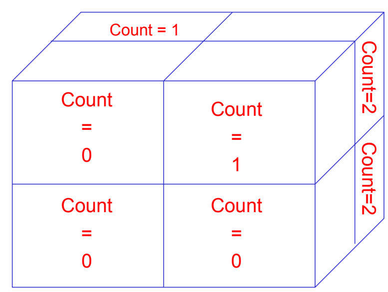
SDSS Galaxies using ADtrees

1,580,000 Galaxies, 27 binary attributes per galaxy.
Time to build ADtree: 4 minutes. Tree Memory: 2 Megs

Operation	Efficient Non-adtree implementation	AD-tree (secs)
All 1-cubes	7 mins	0.1 seconds
All 2-cubes	1 hour, 20 mins	1 second
All 3-cubes	10 hours	11 seconds
All 4-cubes	5 days	1.3 minutes
All 5-cubes	1.2 months	21 minutes
500 iterations of Bayes Net stochastic search	11 minutes	0.2 seconds
50000 iterations of Bayes Net stochastic search	1.3 days	1.9 minutes
Size 2 Association rules	5 minutes	0.05 seconds
Size 4 Association rules	6 hours	6 seconds

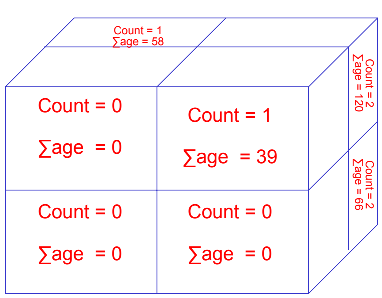
39

Simple Cont Tab



40

Fancy Cont Tab



41

Fancier Cont Tabs

Needs (n , Σ age)

Search for the rule in which	real-valued	output	is maximized
Search for the rule in which	real-valued	output	has least variance
Search for the rule in which	real-valued	output	is closest to target value of 37
Search for the rule in which	categorical-valued	output	has lowest entropy
...and there are dozens of others			

Needs (n_{red} , n_{blue} , n_{green})

Needs (n , Σ age , Σ age^2)

Comparison vs OPUS on finding rules to maximize a real-valued output (comparing against examples published in [Webb, 2001]

	OPUS on a 800 MHz machine	RADSEARCH on a 1.7 GHz machine
covtype/elevation	17 hours	45 secs
ipums/inctot	12 mins	5 mins

Table 4: Comparing RADSEARCH versus real-valued OPUS.

42

Samples of Things you can do with Fancy Cont. Tab Search

- **if** employment=Self ∧ race=White **then** predict age=46.20
- **else if** relation= NonFamily ∧ gender=Female ∧ HoursWorked < 50 **then** predict age=37.11
- **else if...**

Regression List

RADREG:
Regression on propositional terms

- begin with age = 51.6
- **if** marital = NeverMarried subtract 5.09
- **if** edunum > 10 ∧ marital=Married subtract 3.14
- **if** edunum ≤ 10 ∧ marital = NeverMarried ∧ race = White ∧ wealth = poor subtract 3.49
- :

- **if** edunum < 10 ∧ marital=NeverMarried ∧ relation=child **then** predict wealth=poor (99.5% testset agreement)
- **else if** marital=MarriedCivil ∧ job=Professional **then** predict wealth=rich (70.8% testset agreement)
- **else if...**

Decision List

Ole: to our kowledge, OPUS style pruning ot possible for these

43

Summary

	Exact	Approximate is okay
A B C D 1 1 0 1 0 1 1 0 1 0 1 0 0 0 1 1 : : : : — Dense, thin	Dense Datacube	Approx-imation not needed
A B C D ... Z 1 1 0 1 ... 1 0 1 1 0 ... 0 1 0 1 0 ... 1 0 0 1 1 ... 1 : : : : : — Dense, fat	Tree/Hash Datacube, ADtree	Sample, Racing or Approximate ADtree
AA AB AC AD AE AF ... ZZ 0 0 0 1 0 0 ... 1 0 1 0 0 0 0 ... 0 0 0 1 0 0 1 ... 0 0 0 0 0 0 0 ... 0 : : : : : : : — Sparse	Co-occurring tuples	Frequent Sets

44