Sample-Wise Enumeration Methods for Mining Microarray Datasets

Anthony K. H. Tung
Department of Computer Science
National University of Singapore

A Microarray Dataset

1000 - 100,000 columns

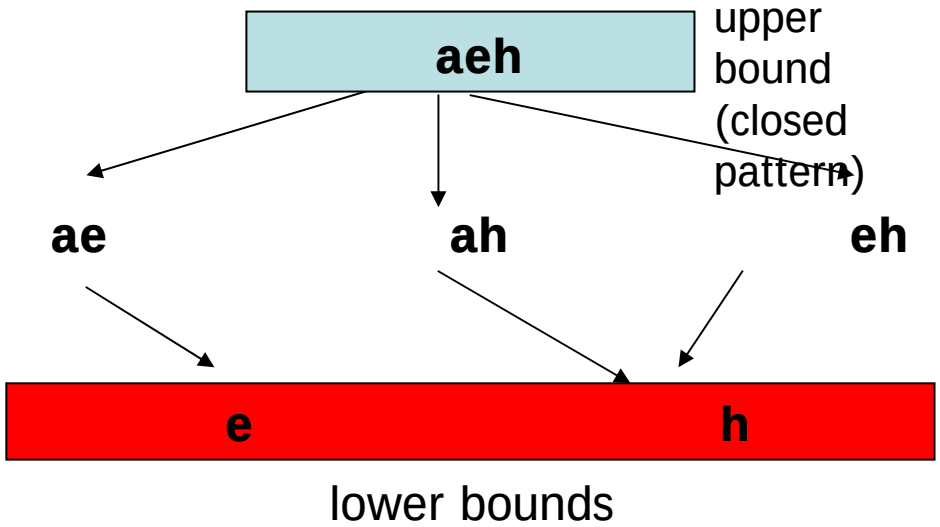
100-500 rows

	Class	Gene1	Gene2	Gene3	Gene4	Gene5	Gene6	Ge
Sample1	Cancer							
Sample2	Cancer							
. . .								
SampleN-1	~Cancer							
SampleN	~Cancer							

- Find closed patterns which occur frequently among genes.
- Find rules which associate certain combination of the columns that affect the class of the rows
  - Gene1,Gene10,Gene1001 -> Cancer

Challenge I

- Large number of patterns/rules
  - number of possible column combinations is extremely high
- Solution: Concept of a closed pattern
  - Patterns are found in exactly the same set of rows are grouped together and represented by their upper bound
- Example: the following patterns are found in row 2,3 and 4



i	*ri*	*Class*
1	*a,b,c,l,o,s*	*C*
2	*a,d, e, h ,p,l,r*	*C*
3	*a,c, e, h ,o,q,t*	*C*
4	*a, e ,f, h ,p,r*	*~C*
5	*b,d,f,g,l,q,s,t*	*~C*

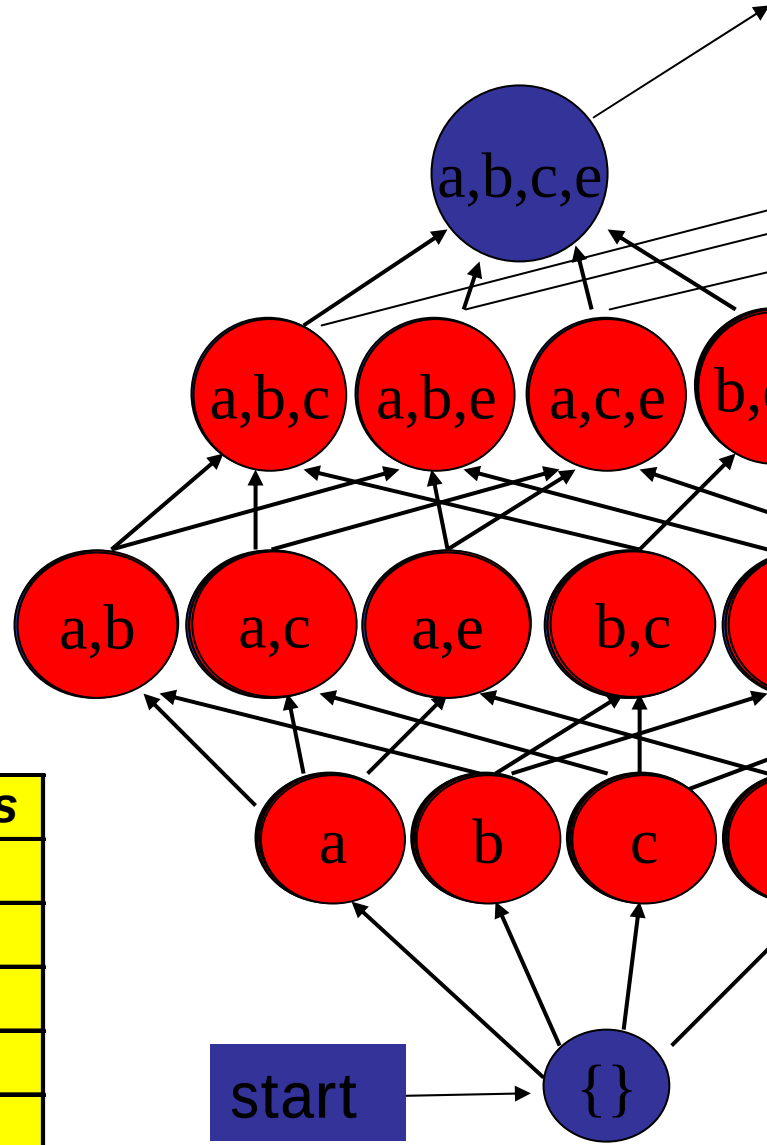
"a" however not part of the group

Challenge II

- Most existing frequent pattern discovery algorithms perform searches in the column/item enumeration space i.e. systematically testing various combination of columns/items
- For datasets with 1000-100,000 columns, this search space is enormous
- Instead we adopt a novel row/sample enumeration algorithm for this purpose. CARPENTER (SIGKDD'03) is the *FIRST* algorithm which adopt this approach

Column/Item Enumeration Lattice

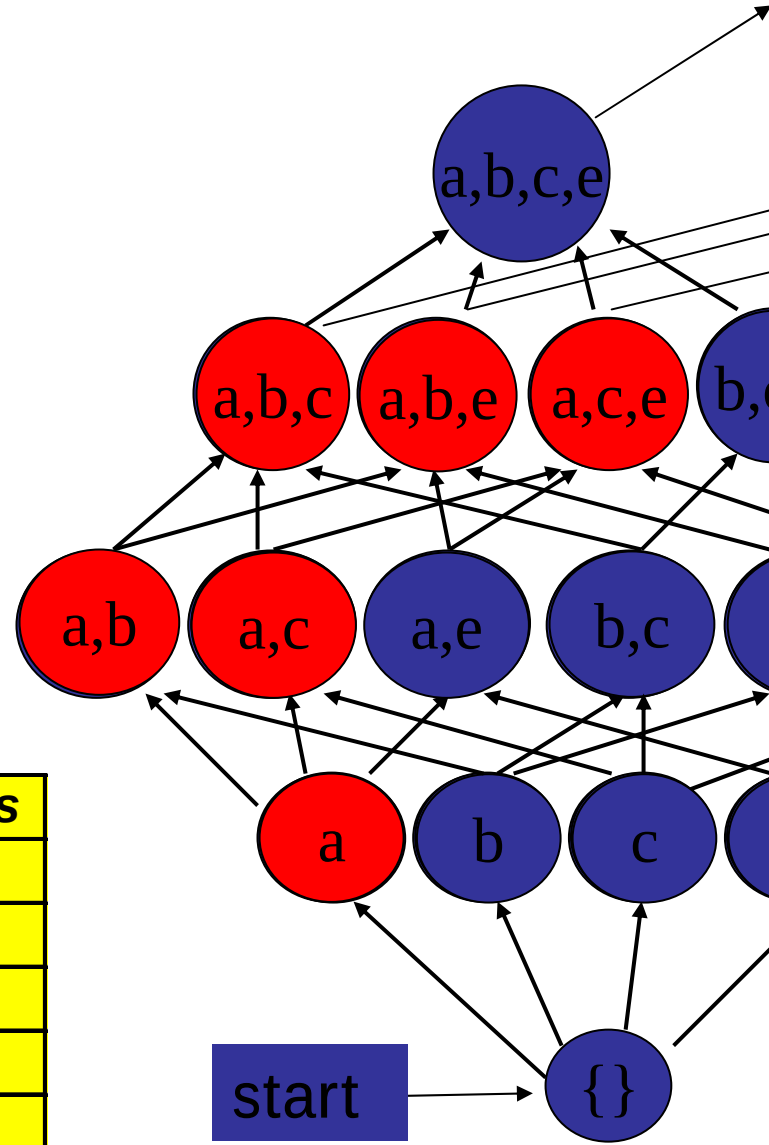
- Each nodes in the lattice represent a combination of columns/items
- An edge exists from node A to B if A is subset of B and A differ from B by only 1 column/item
- Search can be done breadth first



i	*ri*	*Class*
1	*a,b,c,l,o,s*	*C*
2	*a,d,e,h,p,l,r*	*C*
3	*a,c,e,h,o,q,t*	*C*
4	*a,e,f,h,p,r*	*~C*
5	*b,d,f,g,l,q,s,t*	*~C*

Column/Item Enumeration Lattice

- Each nodes in the lattice represent a combination of columns/items
- An edge exists from node A to B if A is subset of B and A differ from B by only 1 column/item
- Search can be done depth first
- Keep edges from parent to child only if child is the prefix of parent

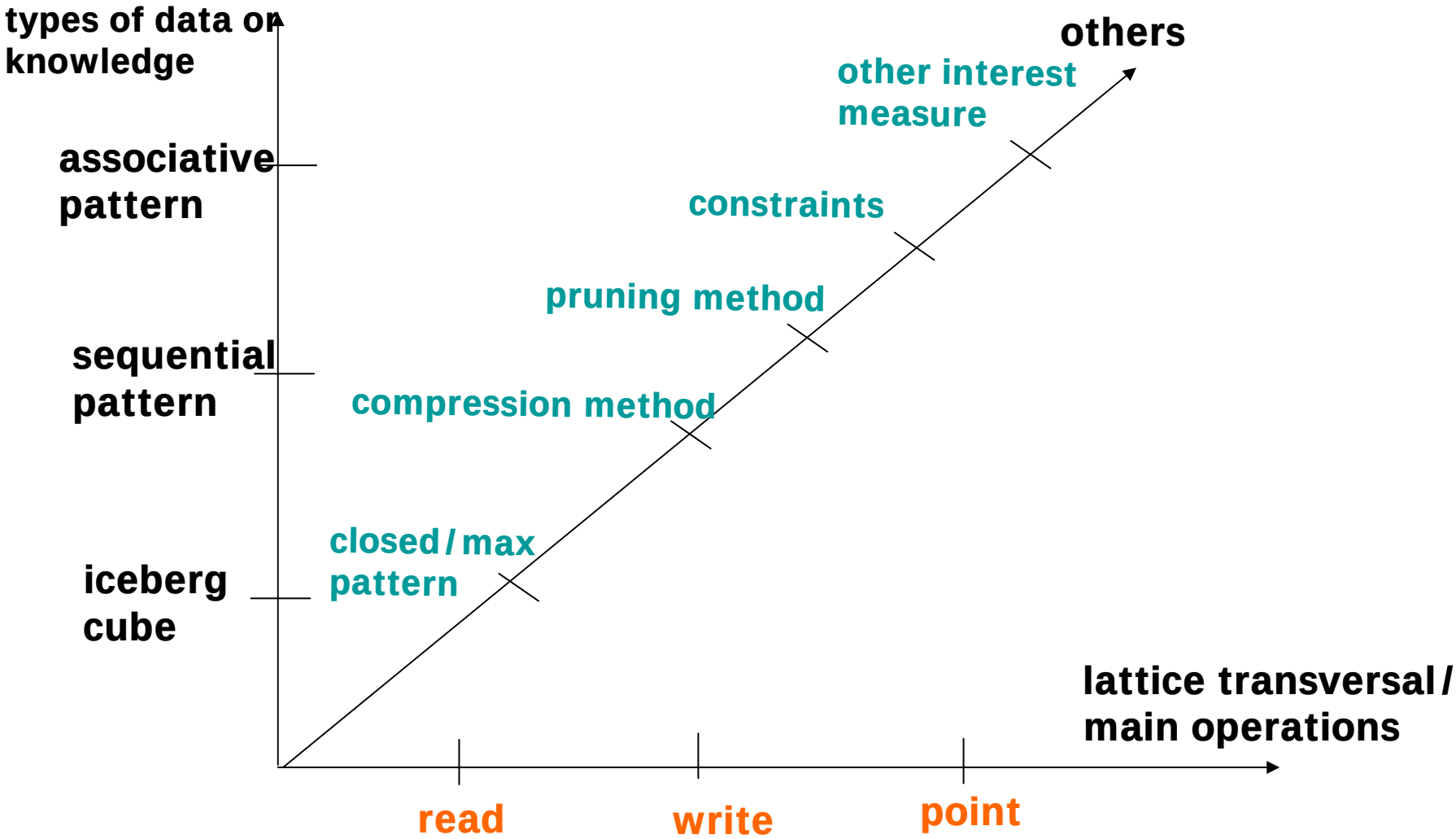


i	*ri*	*Class*
1	*a,b,c,l,o,s*	*C*
2	*a,d,e,h,p,l,r*	*C*
3	*a,c,e,h,o,q,t*	*C*
4	*a,e,f,h,p,r*	*~C*
5	*b,d,f,g,l,q,s,t*	*~C*

General Framework for Column/Item Enumeration

	Read-based	Write-based	Point-based
Association Mining	Apriori[AgSr94], DIC	Eclat, MaxClique[Zaki01], FPGrowth [HaPe00]	Hmine
Sequential Pattern Discovery	GSP[AgSr96]	SPADE [Zaki98,Zaki01], PrefixSpan [PHPC01]	
Iceberg Cube	Apriori[AgSr94]		BUC[BeRa99], H-Cubing [HPDW01]

A Multidimensional View



Sample/Row Enumeration Algorihtms

- To avoid searching the large column/item enumeration space, our mining algorithm search for patterms/rules in the sample/row enumeration space
- Our algorithms does not fitted into the column/item enumeration algorithms
- They are not YAARMA (Yet Another Association Rules Mining Algorithm)
- Column/item enumeration algorithms simply does not scale for microarray datasets

Existing Row/Sample Enumeration Algorithms

- CARPENTER(SIGKDD'03)
  - Find closed patterns using row enumeration
- FARMER(SIGMOD’04)
  - Find interesting rule groups and building classifiers based on them
- COBBLER(SSDBM'04)
  - Combined row and column enumeration for tables with large number of rows and columns
- FARMER's demo (VLDB'04)
- Balance the scale: 3 row enumeration algorithms vs >50 column enumeration algorithms

Concepts of CARPENTER

i	ri	Class
1	a,b,c,l,o,s	C
2	a,d,e,h,p,l,r	C
3	a,c,e,h,o,q,t	C
4	a,e,f,h,p,r	~C
5	b,d,f,g,l,q,s,t	~C

Example Table

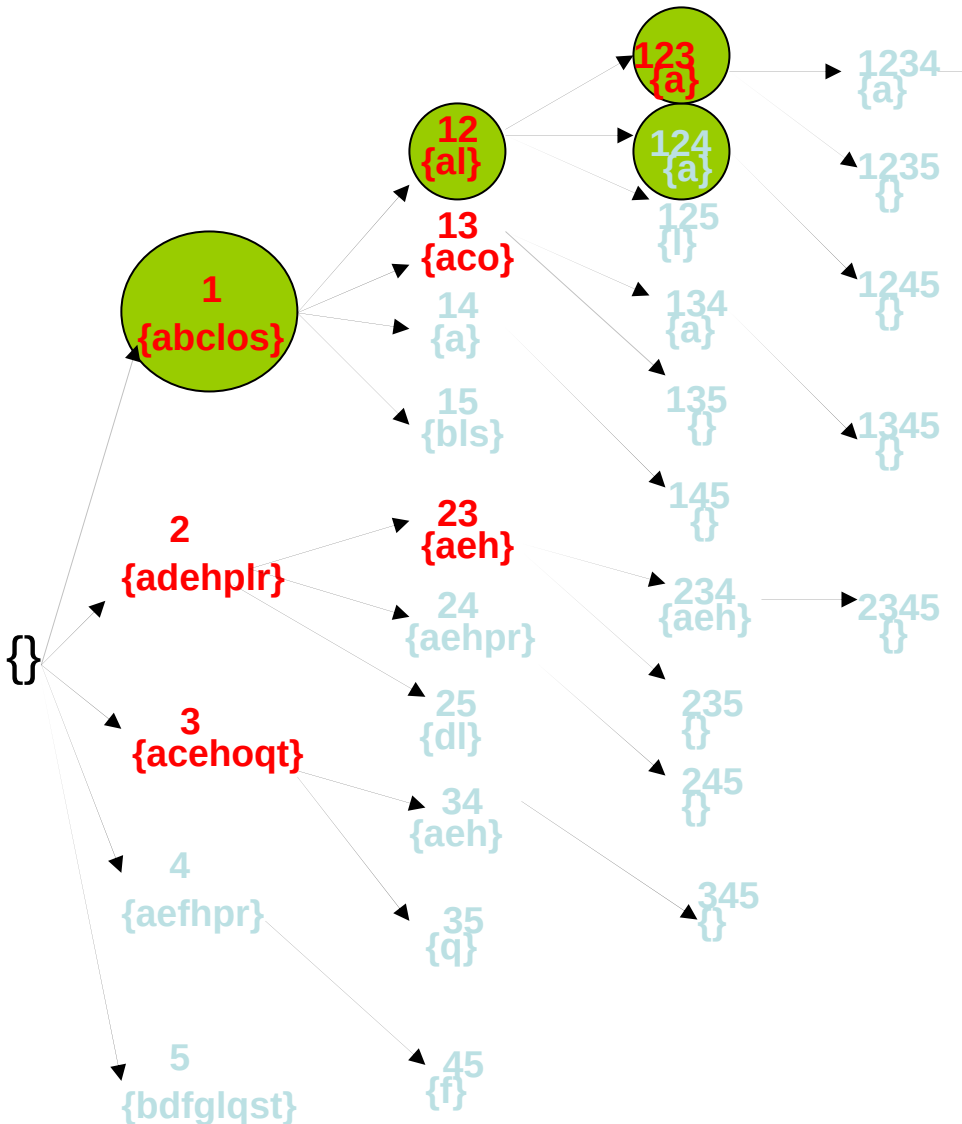
ij	R (ij)	
	C	~C
a	1,2,3	4
b	1	5
c	1,3	
d	2	5
e	2,3	4
f		4,5
g		5
h	2,3	4
l	1,2	5
o	1,3	
p	2	4
q	3	5
r	2	4
s	1	5
t	3	5

Transposed Table,TT

	C	~C
a	1,2,3	4
e	2,3	4
h	2,3	4

$TT|_{\{2,3\}}$

Row Enumeration



$TT|_{\{1\}}$

ij	R (ij)	
	C	~C
a	1,2,3	4
b	1	5
c	1,3	
l	1,2	5
o	1,3	
s	1	5

$TT|_{\{12\}}$

ij	R (ij)	
	C	~C
a	1,2,3	4
l	1,2	5

$TT|_{\{123\}}$

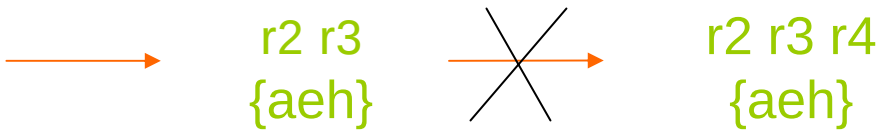
ij	R (ij)	
	C	~C
a	1,2,3	4

ij	R (ij)	
	C	~C
a	1,2,3	4
b	1	5
c	1,3	
d	2	5
e	2,3	4
f		4,5
g		5
h	2,3	4
l	1,2	5
o	1,3	
p	2	4
q	3	5
r	2	4
s	1	5
t	3	5

Pruning Method 1

- Removing rows that appear in all tuples of transposed table will not affect results

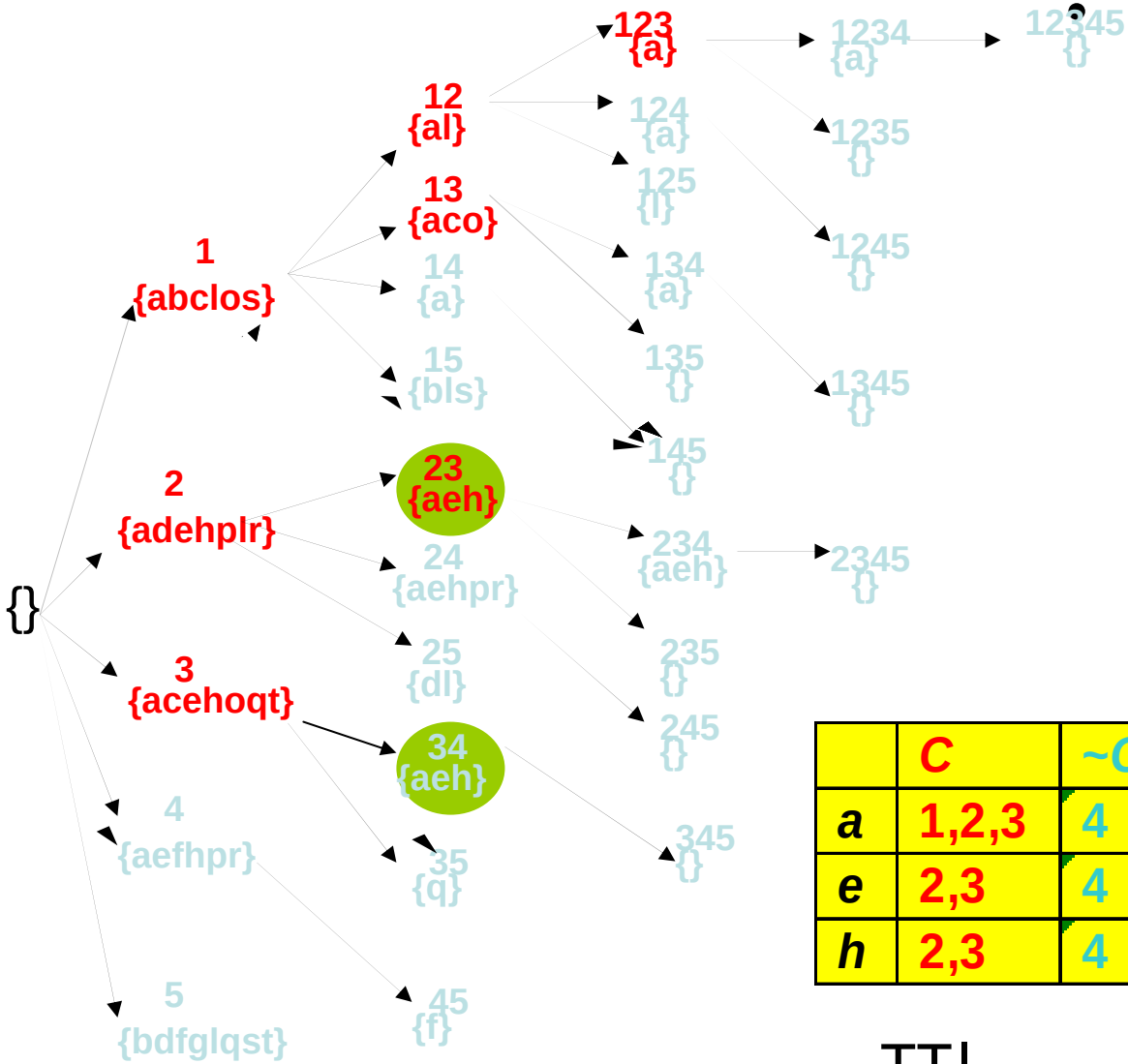
	C	~C
a	1,2,3	4
e	2,3	4
h	2,3	4



$TT|_{\{2,3\}}$

r4 has 100% support in the conditional table of “r2r3”, therefore branch “r2 r3r4” will be pruned.

Pruning method 2



	C	~C
a	1,2,3	4
e	2,3	4
h	2,3	4

$TT|_{\{3,4\}}$

- if a rule is discovered before, we can prune enumeration below this node
  - Because all rules below this node has been discovered before
  - For example, at node 34, if we found that {aeh} has been found, we can prune off all branches below it

Pruning Method 3: Minimum Support

- Example: From $TT|_{\{1\}}$, we can see that the support of all possible pattern below node {1} will be at most 5 rows.

$TT|_{\{1\}}$

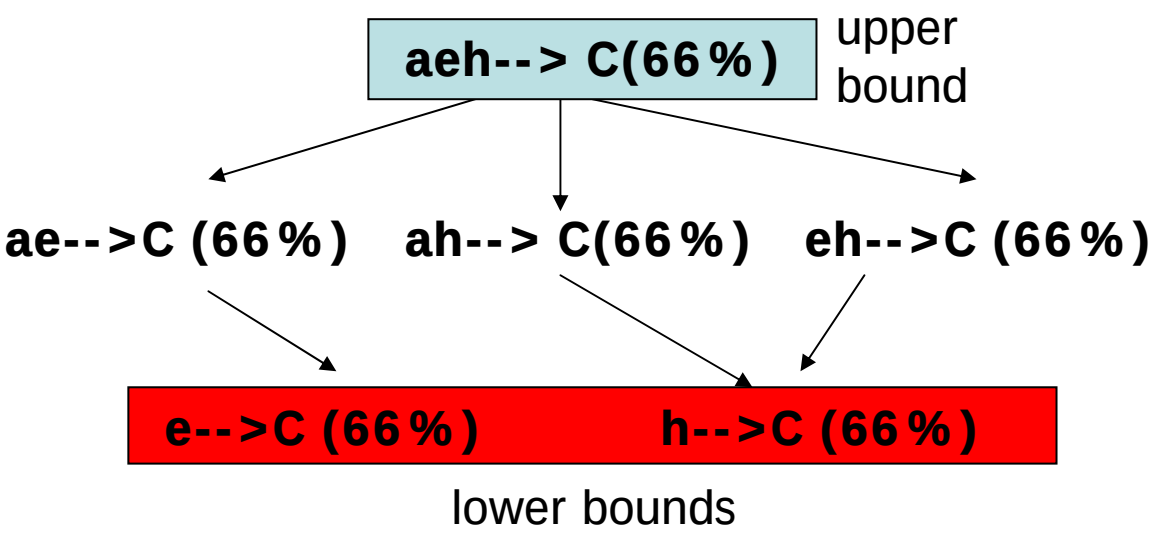
ij	*R*(*ij*)	
	C	~*C*
a	1,2,3	4
b	1	5
c	1,3	
l	1,2	5
o	1,3	
s	1	5

From CARPENTER to FARMER

- What if classes exists ? What more can we do ?
- Pruning with Interestingness Measure
  - Minimum confidence
  - Minimum chi-square
- Generate lower bounds for classification/ prediction

Interesting Rule Groups

- Concept of a rule group/equivalent class
  - rules supported by exactly the same set of rows are grouped together
- Example: the following rules are derived from row 2,3 and 4 with 66% confidence



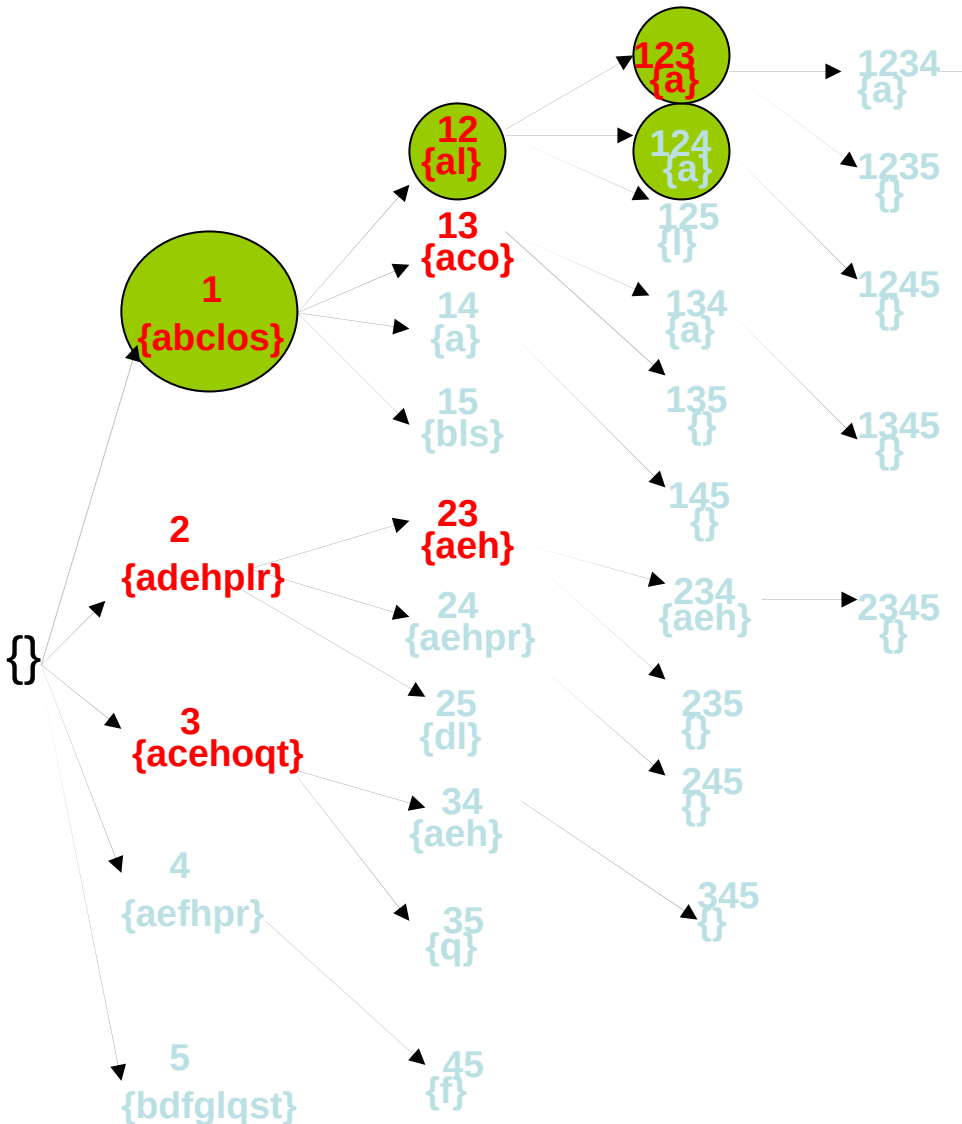
i	ri	Class
1	a,b,c,l,o,s	C
2	a,d,e,h,p,l,r	C
3	a,c,e,h,o,q,t	C
4	a,e,f,h,p,r	~C
5	b,d,f,g,l,q,s,t	~C

a-->C however is not in the group

Pruning by Interestingness Measure

- In addition, find only interesting rule groups (IRGs) based on some measures:
  - minconf: the rules in the rule group can predict the class on the RHS with high confidence
  - minchi: there is high correlation between LHS and RHS of the rules based on chi-square test
- Other measures like lift, entropy gain, conviction etc. can be handle similarly

Ordering of Rows: All Class C before ~C



$TT|_{\{1\}}$

ij	R (ij)	
	C	~C
a	1,2,3	4
b	1	5
c	1,3	
l	1,2	5
o	1,3	
s	1	5

$TT|_{\{12\}}$

ij	R (ij)	
	C	~C
a	1,2,3	4
l	1,2	5

$TT|_{\{123\}}$

ij	R (ij)	
	C	~C
a	1,2,3	4

ij	R (ij)	
	C	~C
a	1,2,3	4
b	1	5
c	1,3	
d	2	5
e	2,3	4
f		4,5
g		5
h	2,3	4
l	1,2	5
o	1,3	
p	2	4
q	3	5
r	2	4
s	1	5
t	3	5

Pruning Method: Minimum Confidence

- Example: In TT|{2,3} on the right, the maximum confidence of all rules below node {2,3} is at most 4/5

	C	*~C*
a	**1,2,3,6**	**4,5**
e	**2,3,7**	**4,9**
h	**2,3**	**4**

$TT|_{\{2,3\}}$

Pruning method: Minimum chi-square

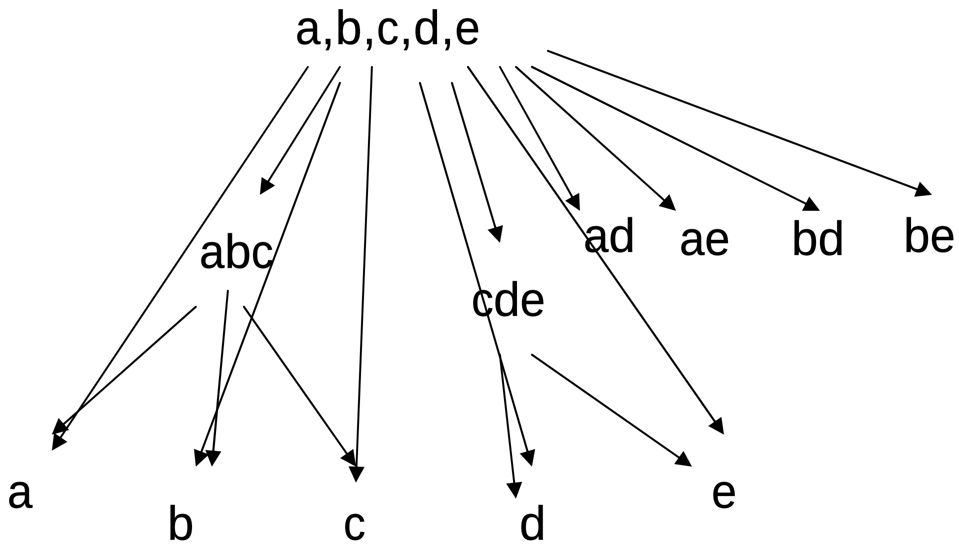
- Same as in computing maximum confidence

	C	~C
a	**1,2,3,6**	**4,5**
e	**2,3,7**	**4,9**
h	**2,3**	**4**

$TT|_{\{2,3\}}$

	C	~C	Total
A	max=5	min=1	Computed
~A	Computed	Computed	Computed
	Constant	Constant	Constant

Finding Lower Bound, MineLB



Candidate lower bound: ad, ae, bd, be, cd, ce

Candidate lower bound: ad, ae, bd, be

Kept since no lower bound override them

Removed since d,e are still lower bound

- Example: An upper bound rule with antecedent A=abcde and two rows (r1 : abcf) and (r2 : cdeg)
- Initialize lower bounds {a, b, c, d, e}
- add “abcf”--- new lower {d ,e}
- Add “cdeg”--- new lower bound{ad, bd, ae, be}

Implementation

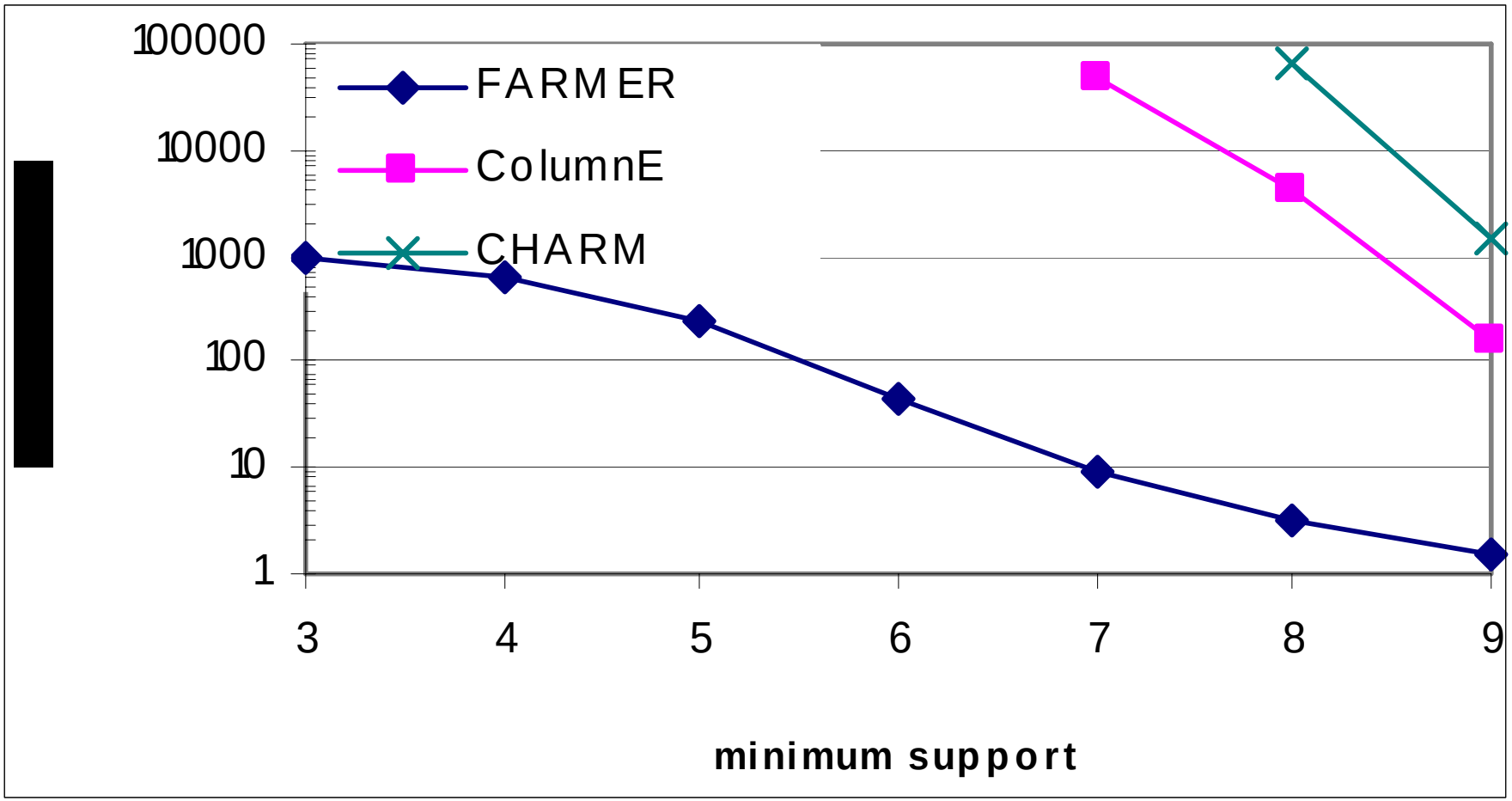
- In general, CARPENTER FARMER can be implemented in many ways:
  - FP-tree
  - Vertical format
- For our case, we assume the dataset can be fitted into the main memory and used pointer-based algorithm similar to BUC

ij	*R* (*ij*)	
	C	~*C*
a	1,2,3	4
b	1	5
c	1,3	
d	2	5
e	2,3	4
f		4,5
g		5
h	2,3	4
l	1,2	5
o	1,3	
p	2	4
q	3	5
r	2	4
s	1	5
t	3	5

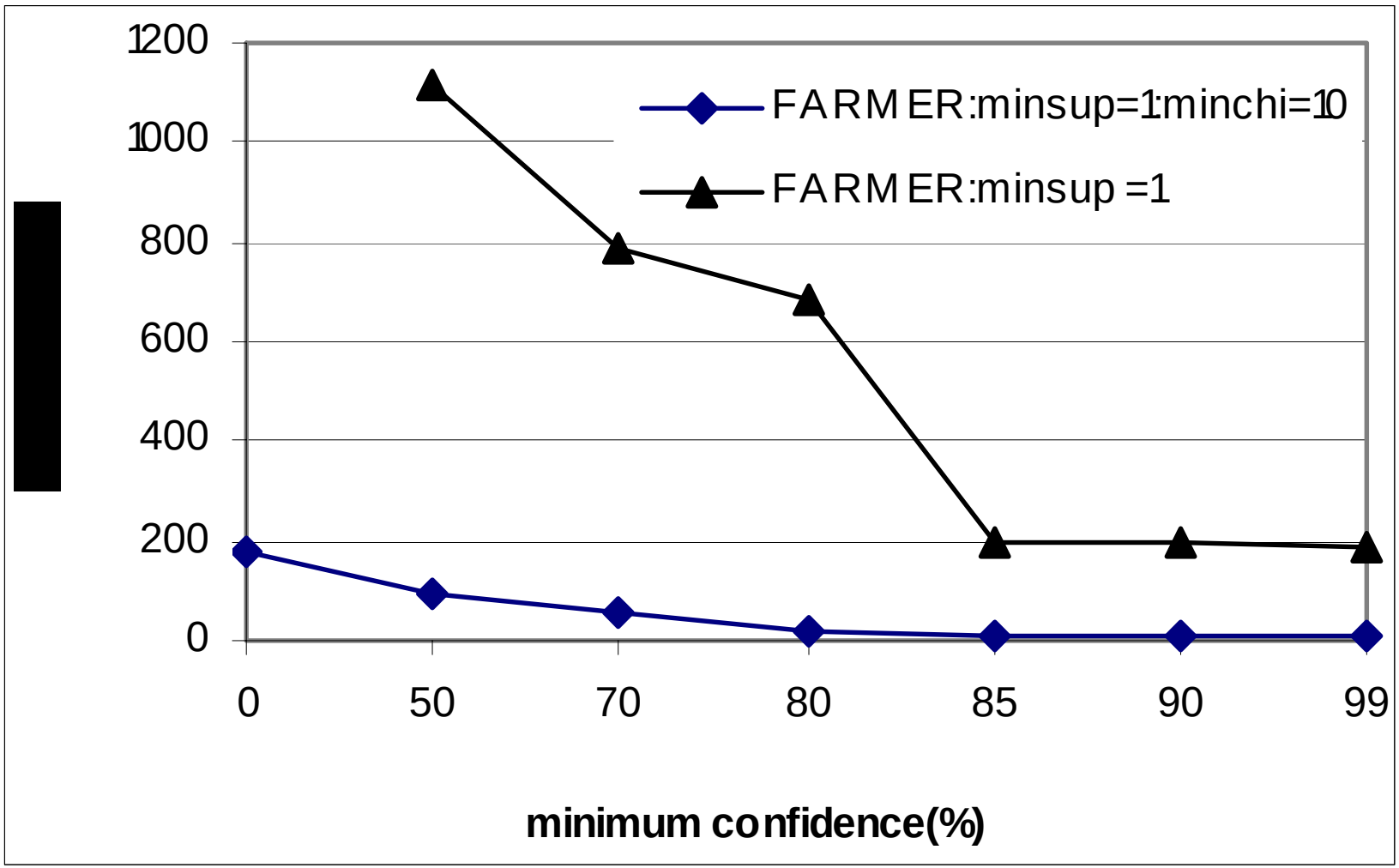
Experimental studies

- Efficiency of FARMER
  - On five real-life dataset
    - lung cancer (LC), breast cancer (BC) , prostate cancer (PC), ALL-AML leukemia (ALL), Colon Tumor(CT)
  - Varying minsup, minconf, minchi
  - Benchmark against
    - CHARM [ZaHs02] ICDM'02
    - Bayardo's algorithm (ColumE) [BaAg99] SIGKDD'99
- Usefulness of IRGs
  - Classification

Example results--Prostate



Example results--Prostate



Naive Classification Approach

- Generate the upper bounds of IRGs
- Rank the upper bounds, thus ranking the IRGs;
- Apply coverage pruning on the IRGs;
- Predict the test data based on the IRGs that it covers.

Classification results

Dataset	# features	# training	# test	IRG Classifier	CBA	SVM
BC	619	78	19	**78.95%**	57.89%	36.84%
LC	2173	32	149	89.93%	81.88%	**96.64%**
CT	135	47	15	**93.33%**	73.33%	73.33%
PC	1554	102	34	**88.24%**	82.35%	79.41%
ALL	866	38	34	64.71%	91.18%	**97.06%**
Average Accuracy				**83.03%**	77.33%	76.66%

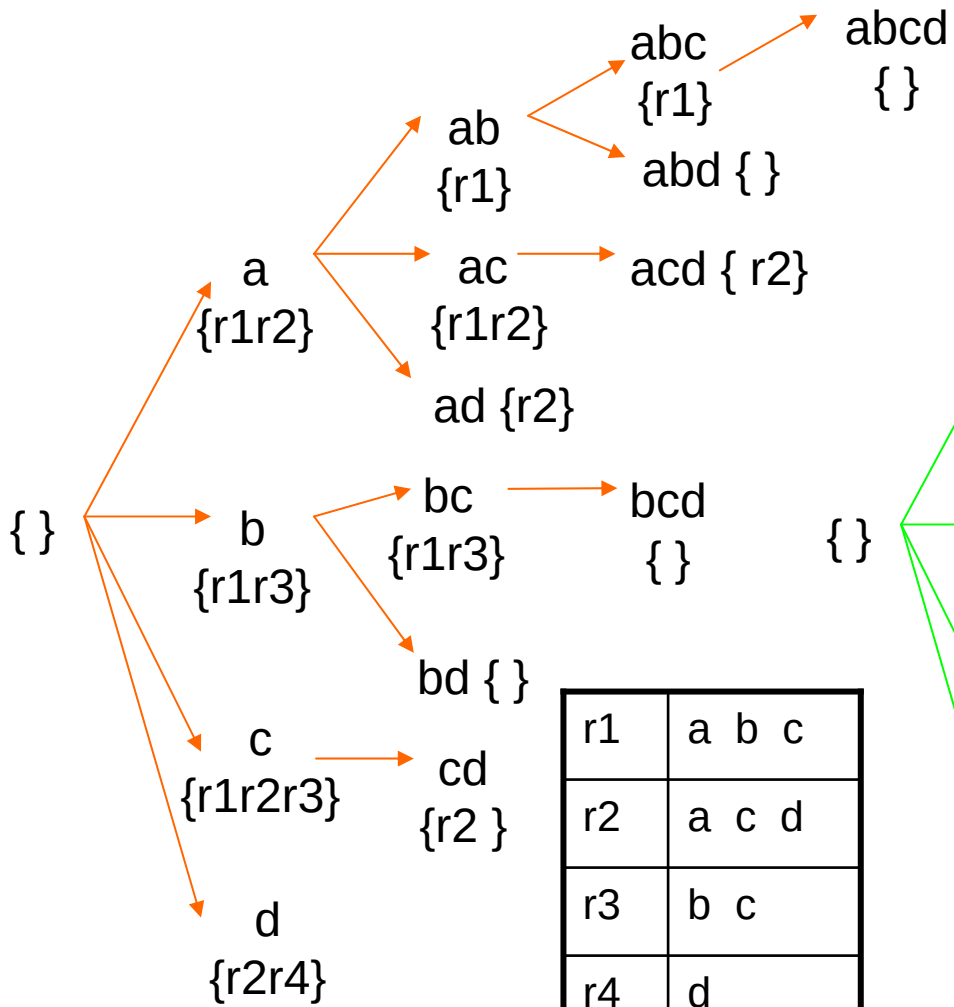
Summary of Experiments

- FARMER is much more efficient than existing algorithms
- There are evidences to show that IRGs is useful for classification of microarray datasets

COBBLER: Combining Column and Row Enumeration

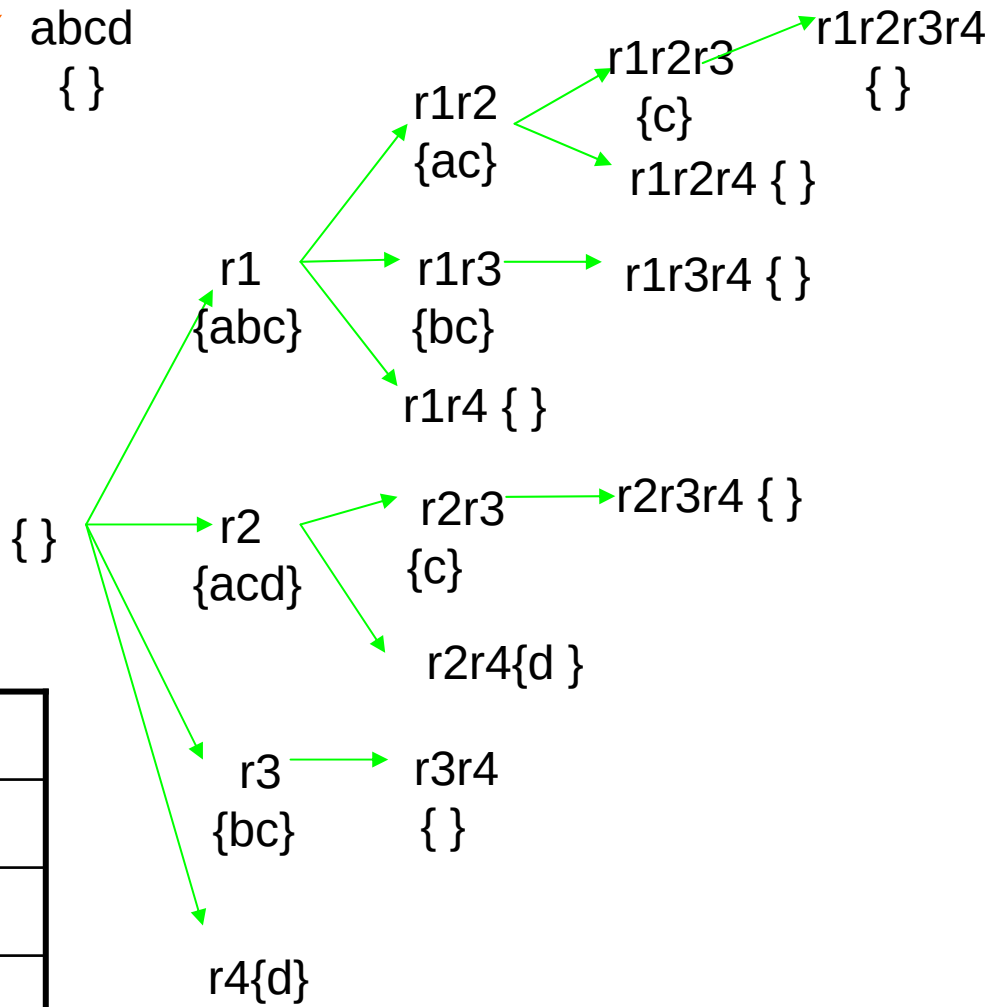
- Extend CARPENTER to handle datasets with both large number of columns and rows
- Switch dynamically between column and row enumeration based on estimated cost of processing

Single Enumeration Tree



r1	a b c
r2	a c d
r3	b c
r4	d

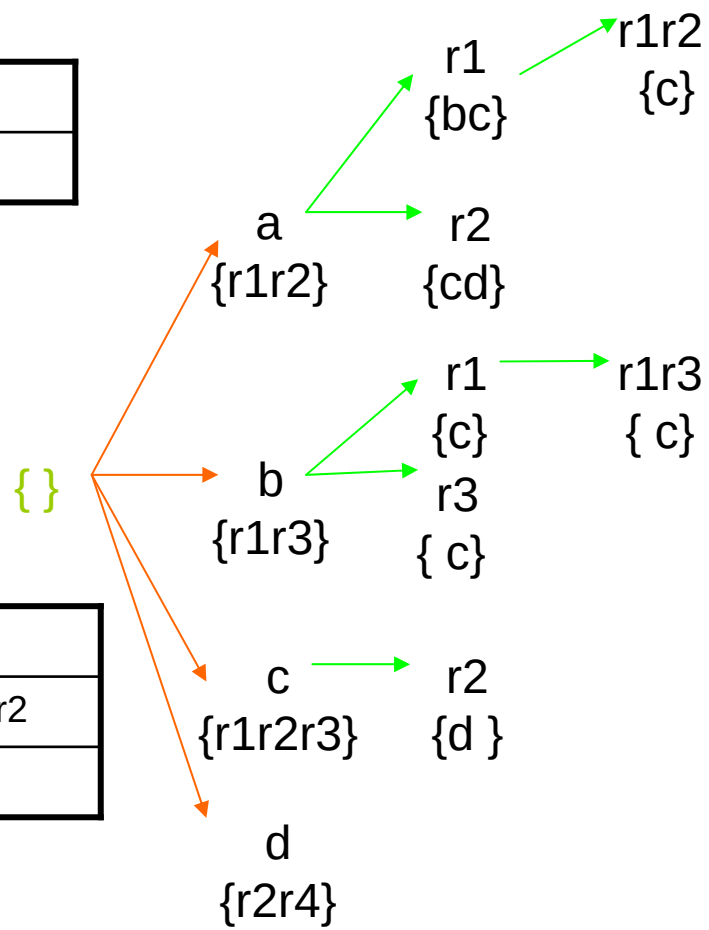
Feature enumeration



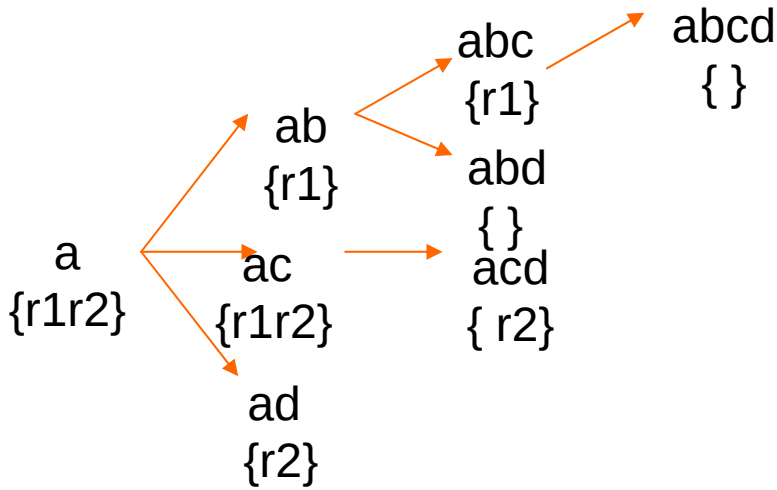
Row enumeration

Dynamic Enumeration Tree

r1	bc
r2	cd



b	r1
c	r1 r2
d	r2



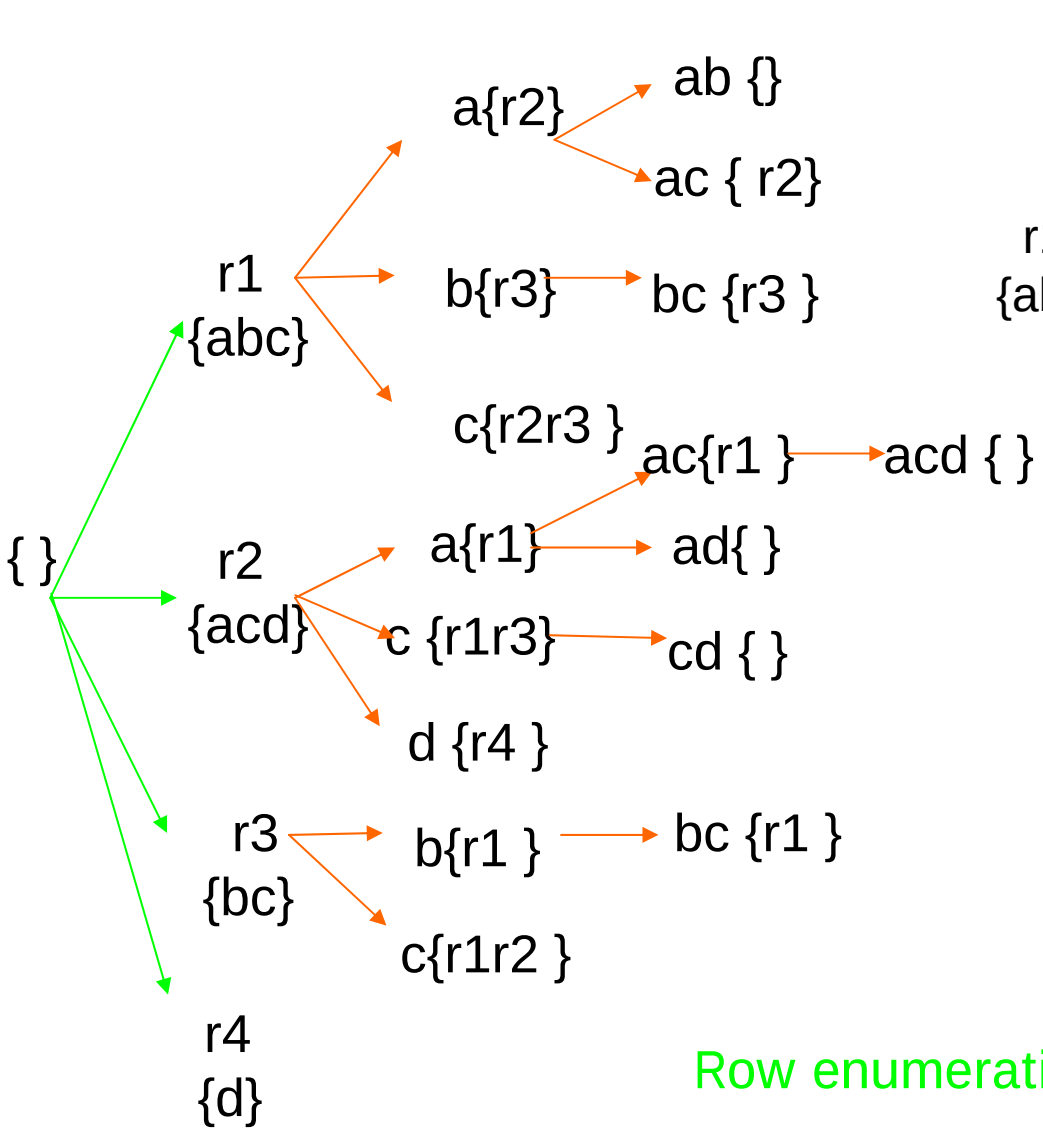
abc: {r1}

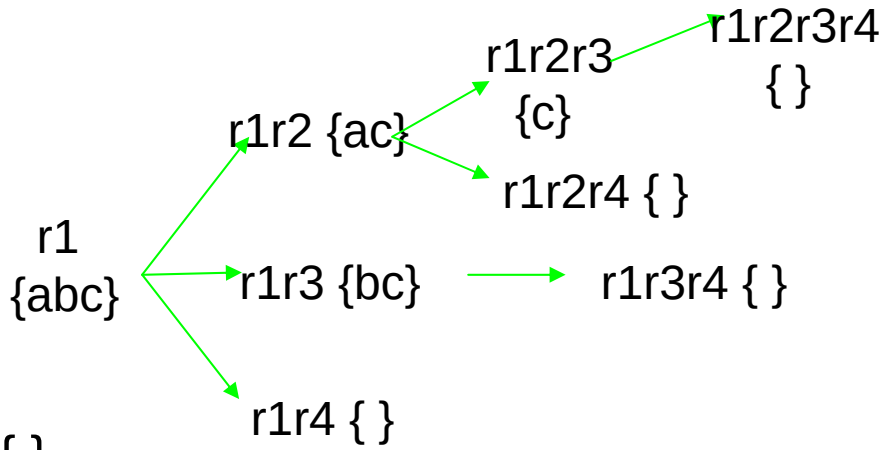
ac: {r1r2}

acd: {r2}

Feature enumeration to Row enumeration

Dynamic Enumeration Tree





ac: {r1r2}

bc: {r1r3}

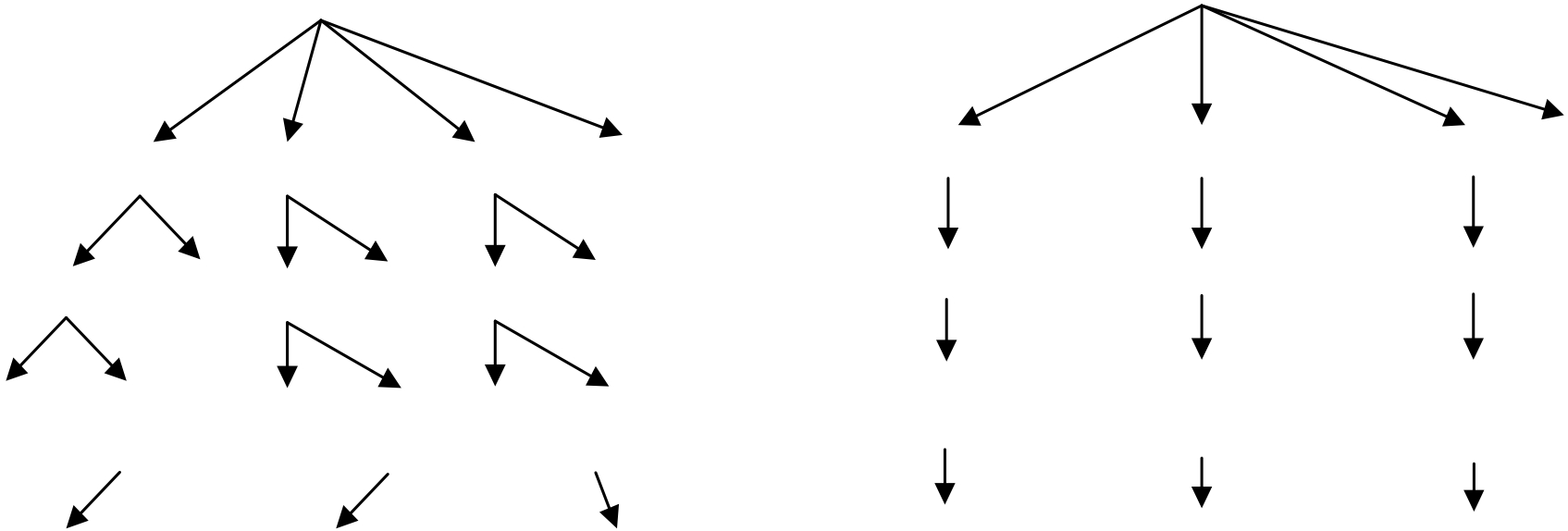
c: {r1r2r3}

Row enumeration to Feature Enumeration

Switching Condition

- Naïve idea of switching based on row number and feature number does not work well
- to estimate the required computation of an enumeration sub-tree, i.e., row enumeration sub-tree or feature enumeration sub-tree.
  - Estimate the maximal level of enumeration for each children sub-tree
- Example of estimating the maximal level of enumeration:
  - Suppose r=10, S(f1)=0.8, S(f2)=0.5, S(f3)=0.5, S(f4)=0.3 and *minsup*=2
  - S(f1)*S(f2)*S(f3)*r =2 $\geqslant$ *minsup*
  - S(f1)*S(f2)*S(f3)*S(f4)*r =0.6 < *minsup*
  - Then the estimated deepest node under f1 is f1f2f3

Switching Condition



f1 f2 f3 … fn

{f1,f2}..{f1,fn} {f2,f3}..{f2,fn} {f3,f4}..{f3,fn}

Switching Condition

- $F = \{f_1, f_2, \ldots, f_n\}$.
- $\mathcal{S}(f_j, T|_F)$, the frequency of feature f_j in $T|_F$.
- $r = |\mathcal{R}(F)|$, the number of rows conditional table $T|_F$ contains.

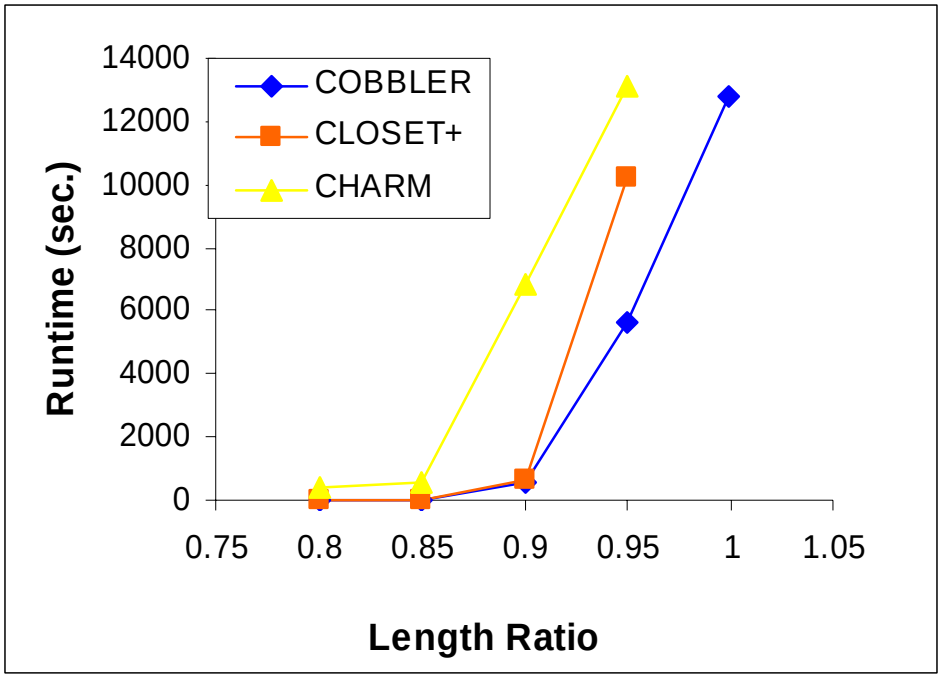
To estimate for a node:

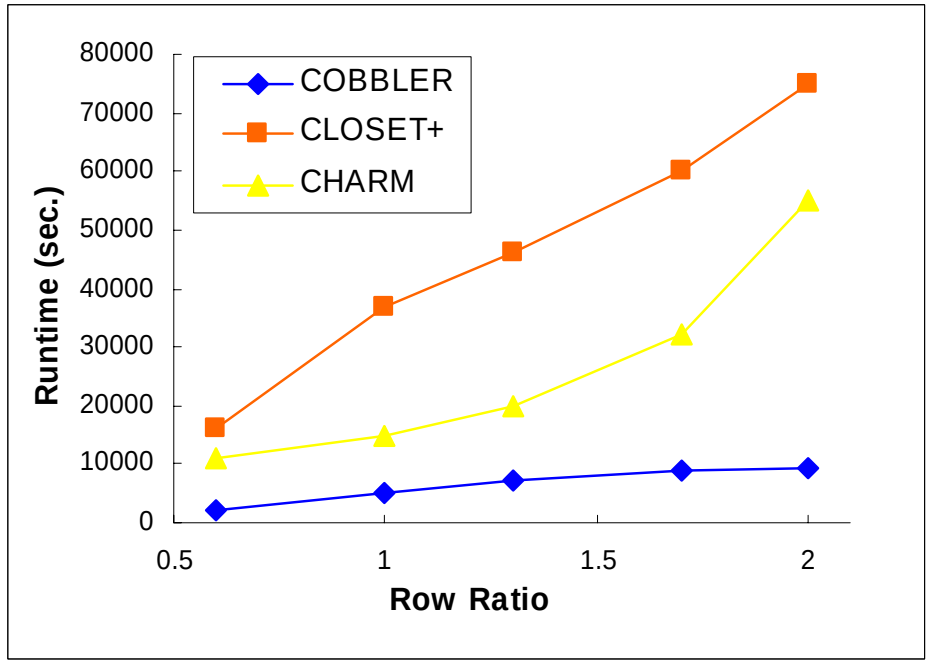
To estimate for a path:

$$\mathcal{L}(N_i) = \sum_{k=1}^{h} (r \cdot RowProcessTime \cdot \prod_{j=1}^{k} \mathcal{S}(f_j, T|_{F_i'})).$$

To sum up estimation of all paths as the final estimation

Length and Row ratio





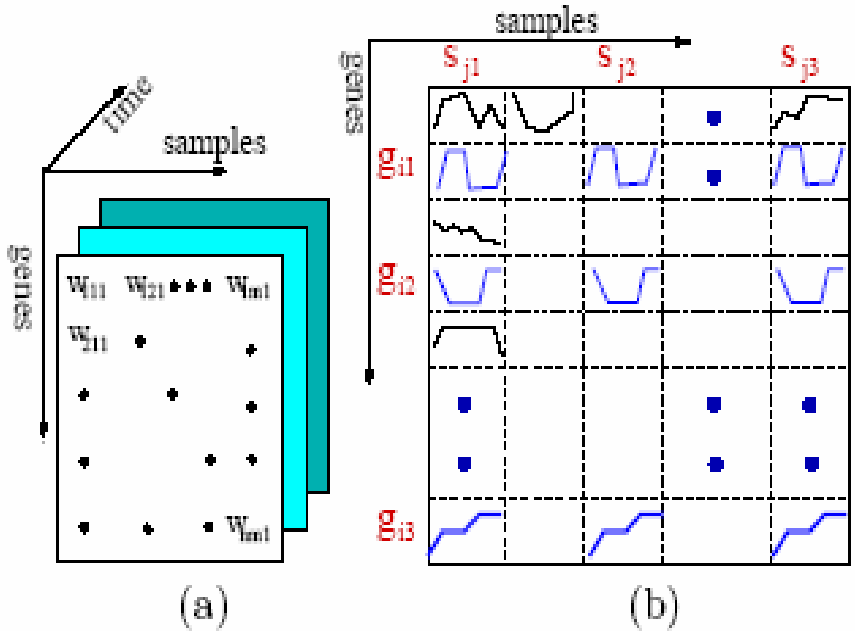
Synthetic data

Extension of our work by other groups (with or without citation)

- *[1]* *Using transposition for pattern discovery from microarray data*, Francois Rioult (GREYC CNRS), Jean-Francois Boulicaut (INSA Lyon), Bruno Cremileux (GREYC CNRS), Jeremy Besson (INSA Lyon)
- See the presence and absence of genes in the sample as a binary matrix. Perform a transposition of the matrix which is essentially our transposed table. Enumeration methods are the same otherwise.

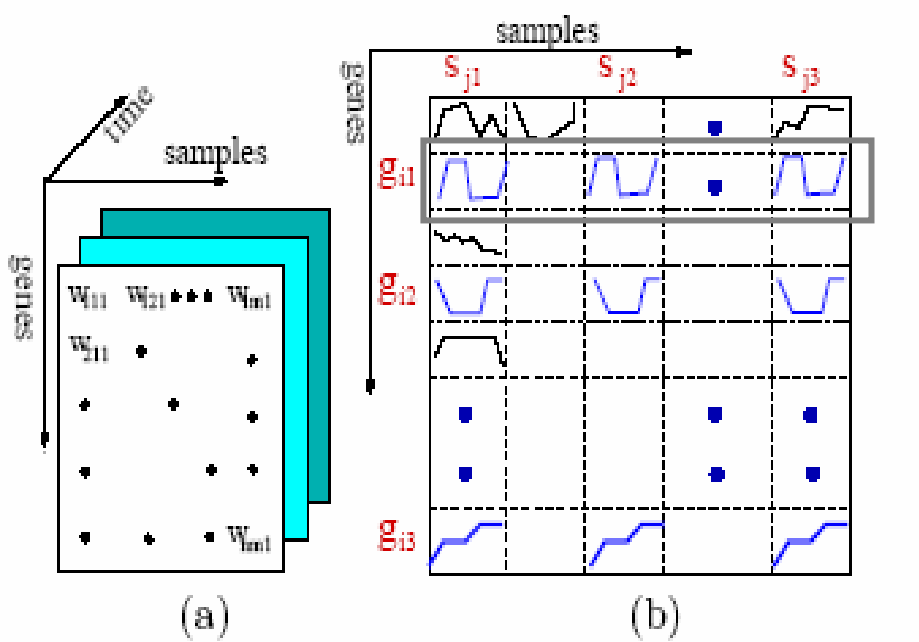
Extension of our work by other groups (with or without citation) II

- [2] Mining Coherent Gene Clusters from Gene-Sample-Time Microarray Data. D. Jiang, **Jian Pei**, M. Ramanathan, C. Tang and A. Zhang. (**Industrial full paper**, Runner-up for the best application paper award). *SIGKDD'2004*



	Gene1	Gene 2	Gene3	Gene 4
Sample1				
Sample2				
. . .				
SampleN-1				
SampleN				

Extension of our work by other groups (with or without citation) III



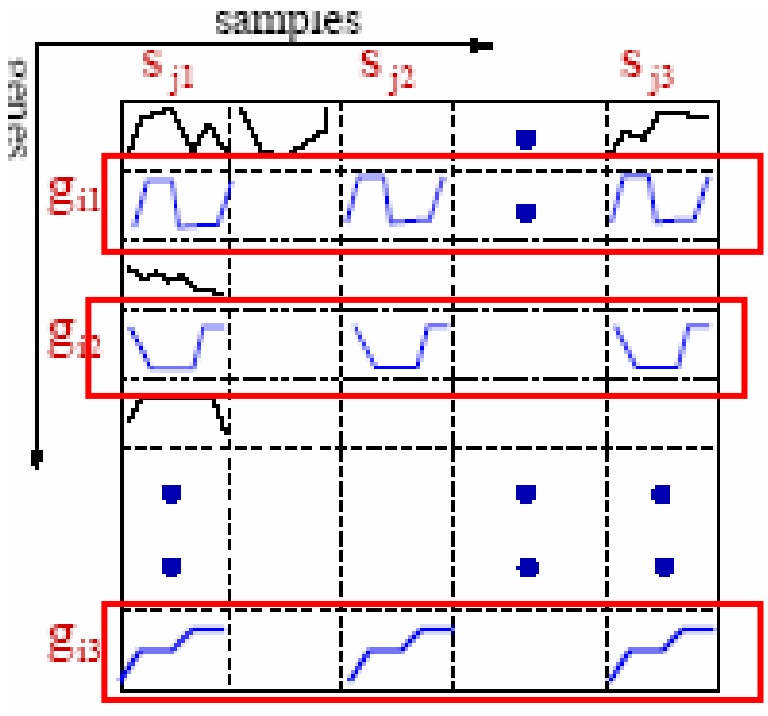
	Gene 1	Gene 2	Gene 3	Gen 4
S1		1.23		
S2		1.34		
. . .				
SN-1		1.52		
SN				

A gene in two samples are say to be coherent if their time series satisfied a certain matching condition

In CARPENTER, a gene in two samples are say to be matching if their expression in the two samples are almost the same

Extension of our work by other groups (with or without citation) IV

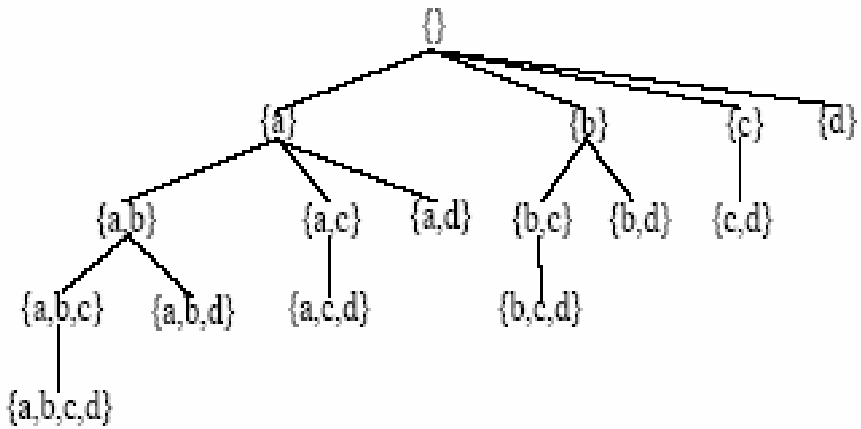
[2] Try to find a subset of samples S such that a subset of genes G is coherent for each pair of samples in S.
$|S|>min_s$, $|G|>min_g$



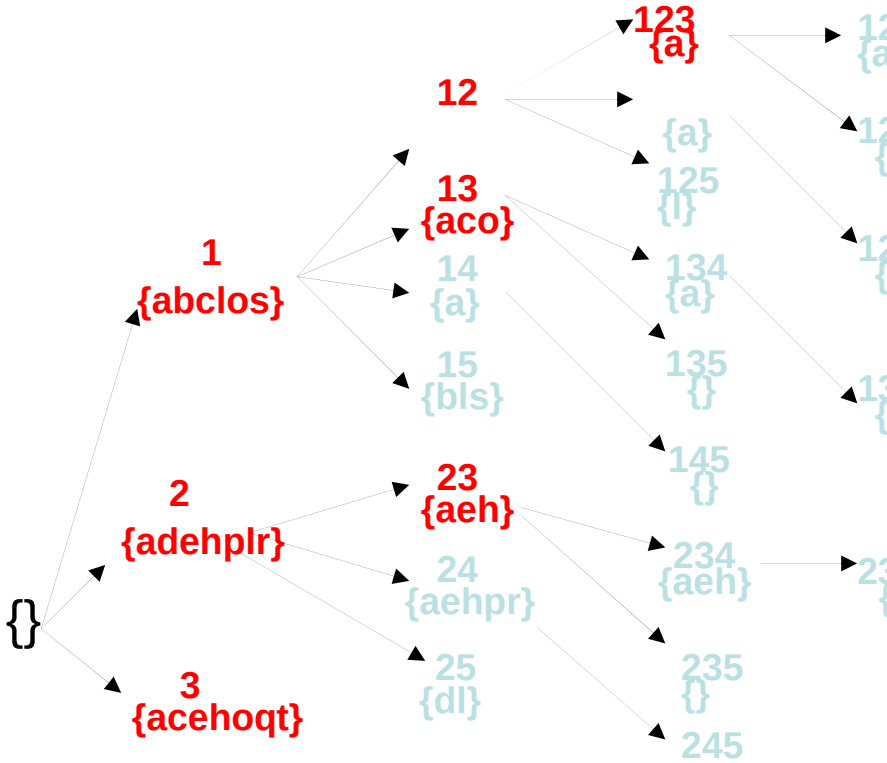
In CARPENTER, we try to find a subset of samples S in which a subset of genes G is similar in expression level for each pair of samples in S.
$|S|>min_s$, $|G|>0$

	Gene1	Gene2	Gene 3	Gene4
S1		1.23		
S2		1.34		
. . .				
SN-1		1.52		
SN				

Extension of our work by other groups (with or without citation) V



[2] Perform sample-wise enumeration and remove genes that are not pairwise coherent across the samples enumerated



CARPENTER: Perform sample-wise enumeration and remove genes that does not have the same expression level across the samples enumerated

Extension of our work by other groups (with or without citation) VI

From [2]: Pruning Rule 3.1 (Pruning small sample sets). *At a node v = fsi1 ; : : : ; sikg, the subtree of v can be pruned if* (*k* + *jTailj*) < min_s

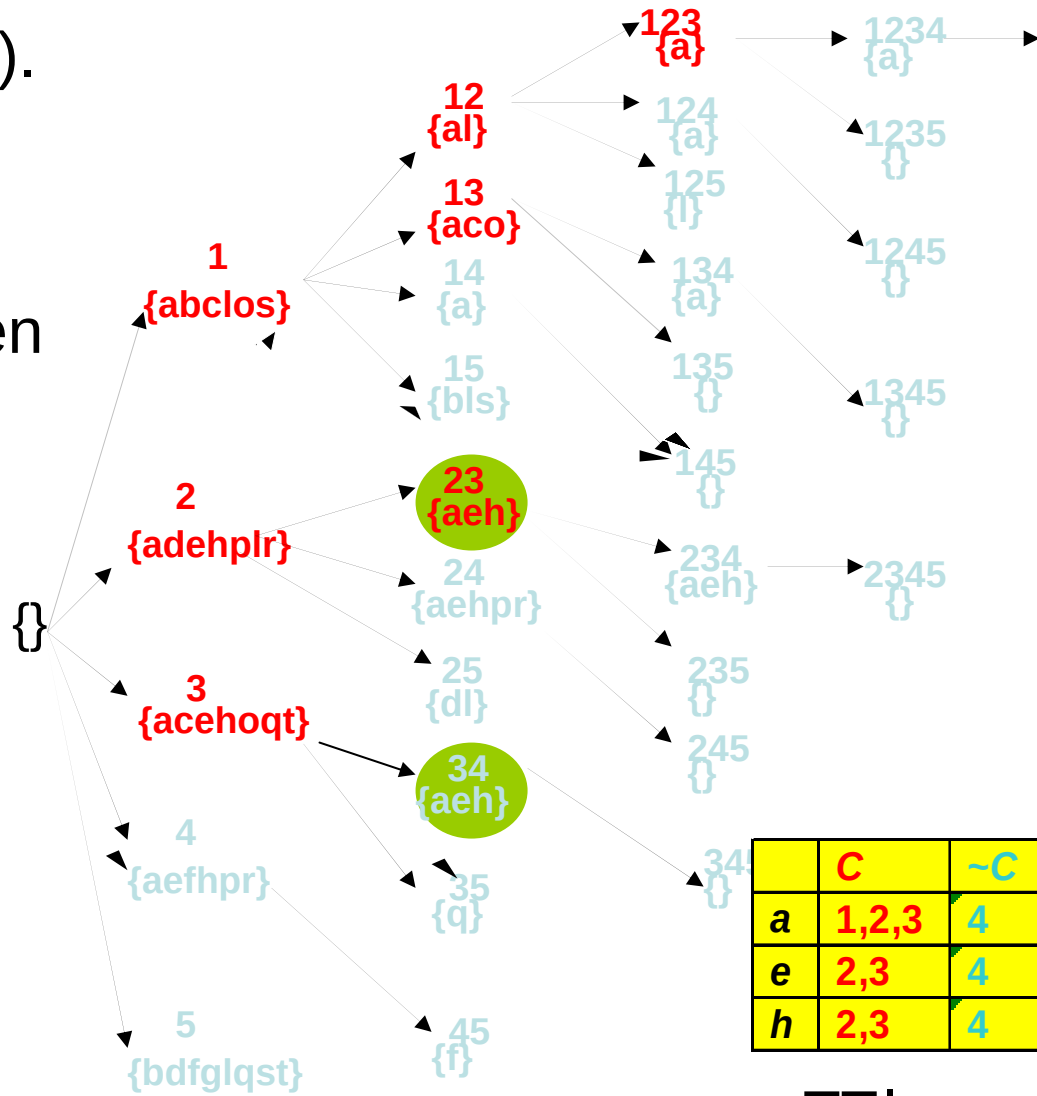
$TT|_{\{1\}}$

ij	R(ij)	
	C	~C
a	1,2,3	4
b	1	5
c	1,3	
l	1,2	5
o	1,3	
s	1	5

- Pruning Method 3 in CARPENTER: From $TT|_{\{1\}}$, we can see that the support of all possible pattern below node {1} will be at most 5 rows.

Extension of our work by other groups (with or without citation) VII

- [2] Pruning Rule 3.2 (Pruning subsumed sets). *At a node* $v = \{s_i \ldots s_{ik}\}$ *if* $\{si1,\ldots s_{ik}\}$ *U Tail* is a subset of some maximal coherent sample set, then the subtree of the node can be pruned.



	C	~C
a	1,2,3	4
e	2,3	4
h	2,3	4

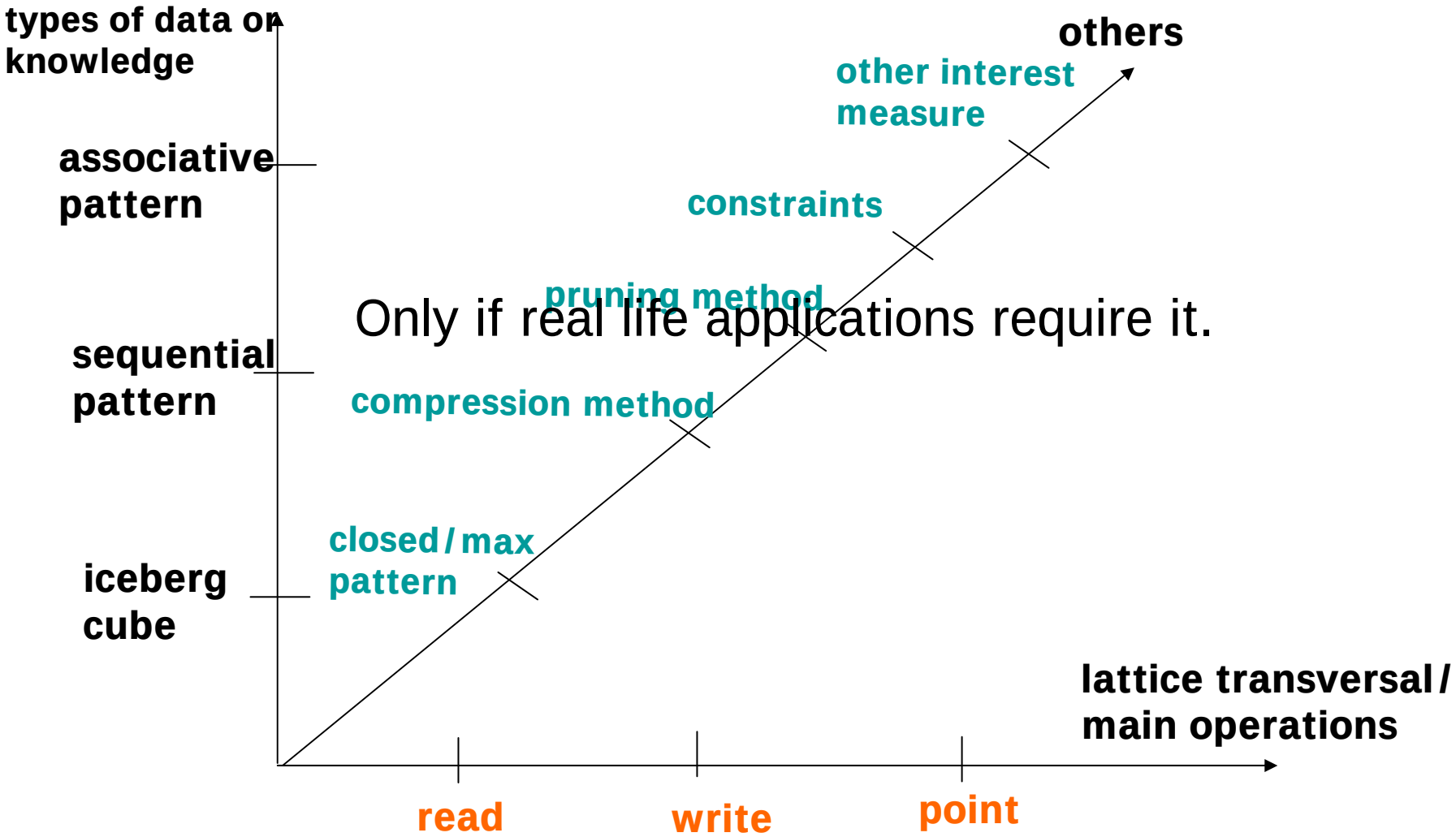
$TT|_{\{3,4\}}$

- CARPENTER Pruning Method 2: if a rule is discovered before, we can prune enumeration below this node

Extension of our work (Conclusion)

- The sample/enumeration framework had been successfully adopted by other groups in mining microarray datasets
- We are proud of our contribution as the group the produce the first row/sample enumeration algorithm CARPENTER and is happy that other groups also find the method useful
- However, citations from these groups would have been nice. After all academic integrity is the most important things for a researcher.

Future Work: Generalize Framework for Row Enumeration Algorithms?



Conclusions

- Many datasets in bioinformatics have very different characteristics compared to those that has been previously studied
- These characteristics can either work against you or for you
- In the case of microarray datasets with large number columns but small number of rows/samples, we turn what is against us to our advantage
  - Row/Sample enumeration
  - Pruning strategy
- We show how our methods have been modified by other groups to produce useful algorithm for mining microarray datasets

Thank you!!!

atung@comp.nus.edu.sg

www.comp.nus.edu.sg/~atung/sfu_talk.pdf