CARPENTER
Find Closed Patterns in Long Biological Datasets

Zhiyu Wang

Knowledge Discovery and Data Mining

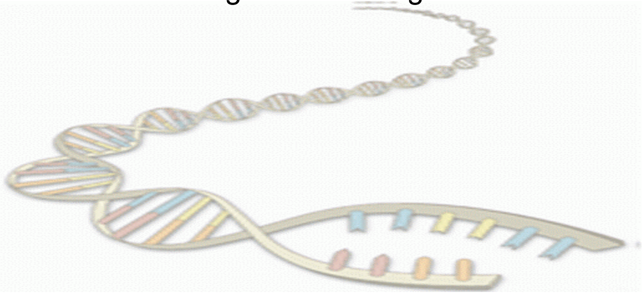
Dr. Osmar Zaiane

Department of Computing Science

University of Alberta

Biological Datasets

- Gene expression
  - Consists of large number of genes



Biological Datasets

- Lung Cancer dataset (gene expression)
  - 181 samples
  - Each sample is described by 12533 genes

How can we find frequent patterns in such dataset?

CARPENTER

Overview……

- **Motivation**
- **Problem statement**
- **Preliminaries**
- **CARPENTER algorithm**
  - **Transpose table**
  - **Row enumeration tree**
  - **Prune methods**
- **Performance**
- **Comments and Conclusion**

Motivation

- **Challenge to find the closed patterns from biological datasets that contains large number of columns with small number of rows**
  - For example,
    10,000 – 100,000 columns with 100 – 1,000 rows

Motivation

- **Running time of most existing algorithms increases exponentially with increasing average row length**
  - For example, in a dataset potential 2^i frequent itemsets, where i is the maximum row size.
  - What if i=12533?

$$2^{12533} = 6.44 \times 10^{3772}$$ (Hugh Search Space)

Problem Statement

- Discover all the frequent closed patterns with respect to user specified support threshold in such biological datasets efficiently.

Preliminaries

- Features f_i
  - Items in the dataset
- Feature support set $R(F')$
  - Maximal set of rows contain a set of features F'

i	*r_i*
1	a, b, c
2	b, c, d
3	b, c, d
4	d

Features: {a, b, c, d}

Feature support set
F'={b,c}, then $R(F')$ ={1,2,3}

Preliminaries

- Row support set $F(R')$
  - Maximal set of features common to a set of rows
- Frequent closed pattern
  - There is no superset with the same support value R'

i	*r_i*
1	a, b, c
2	b, c, d
3	b, c, d
4	d

Row support set

R'={1,2}, then $F(R')$={b,c}

Frequent Closed patterns:

{b,c}, {d}, {b,c,d}……..

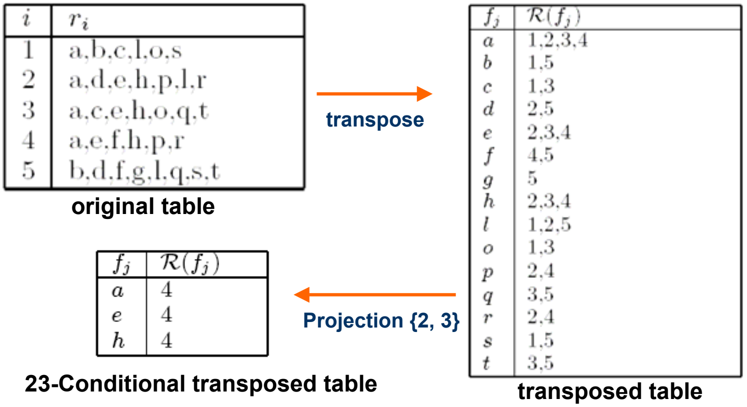
CARPENTER algorithm

- **Proposed by A. K. H. Tung et.al, in ACM SIGKDD 2003.**
- **Main idea is to find frequent closed pattern in depth-first row-wise enumeration.**
- **Assumption: Assume dataset satisfies the condition:** $|R| << |F|$

CARPENTER

- **There are two phases:**
  1. **Transpose the dataset**
  2. **Row enumeration tree**
     - **Recursively search in conditional transposed table**

Transpose table



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

original table

f_j	$\mathcal{R}(f_j)$
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

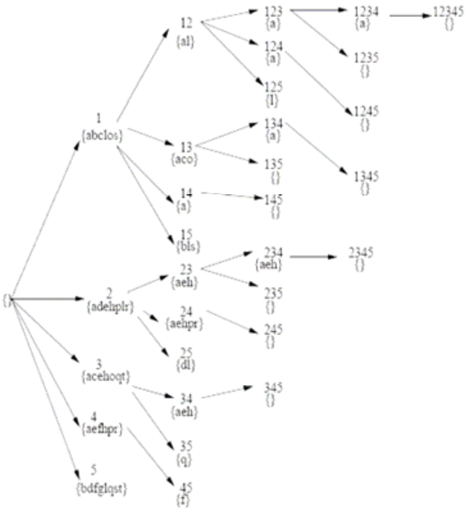
f_j	$\mathcal{R}(f_j)$
a	4
e	4
h	4

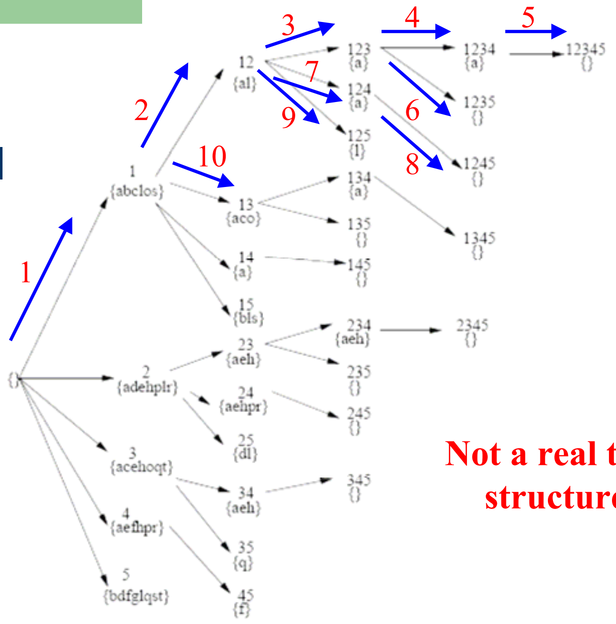
23-Conditional transposed table

Row enumeration tree

- Bottom-up row enumeration tree is based on conditional table.
- Each node is a conditional table.
  - 23-conditional table represents node 23.

f_j	$\mathcal{R}(f_j)$
a	4
e	4
h	4





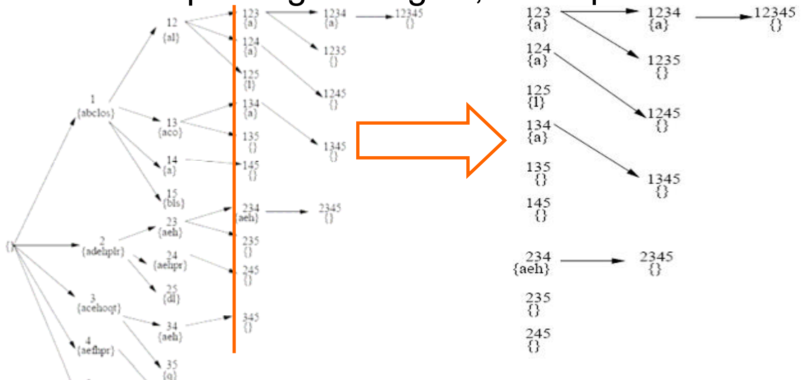
Not a real tree structure

CARPENTER

- Recursively generation of conditional transposed table, performing a depth-first traversal of row-enumeration tree in order to find the frequent closed patterns.

Example

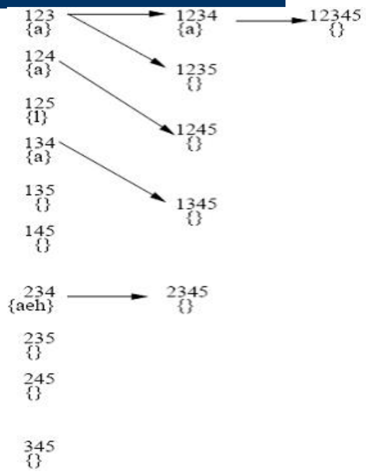
- Without pruning strategies, minsup=3



Example

- Frequent closed patterns

Minsup=3	
a	1,2,3,4
l	1,2,5
aeh	2,3,4

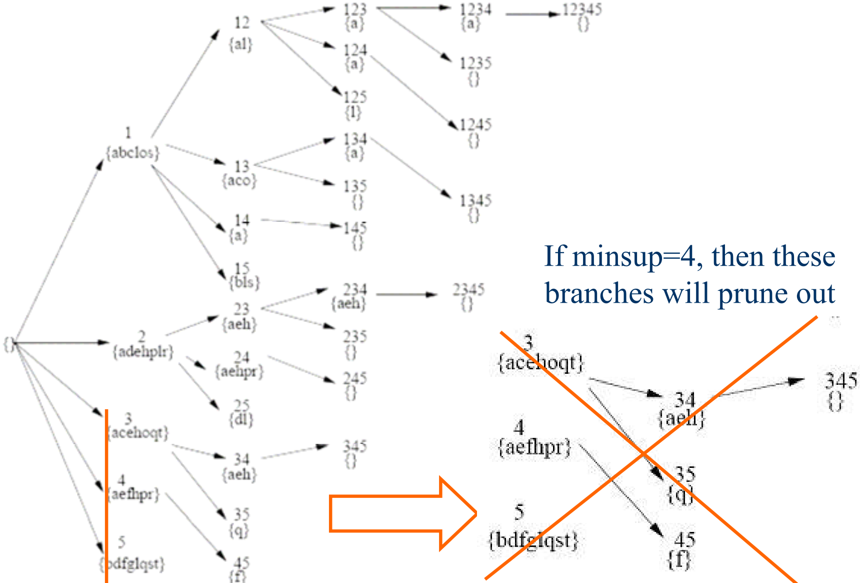


Prune methods

- It is obvious that complete traversal of row enumerations tree is not efficient.
- CARPENTER proposes 3 prune methods.

Prune method 1

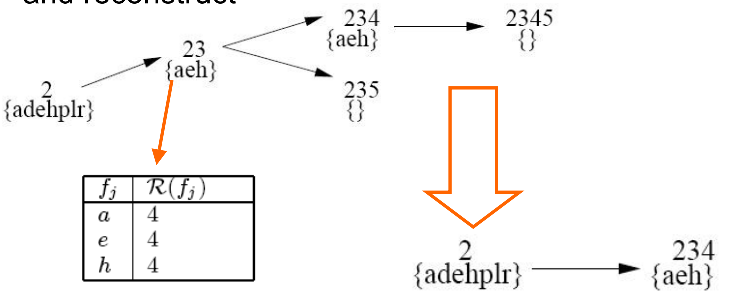
- Prune out the branch which can never generate closed pattern over minsup threshold



If minsup=4, then these branches will prune out

Prune method 2

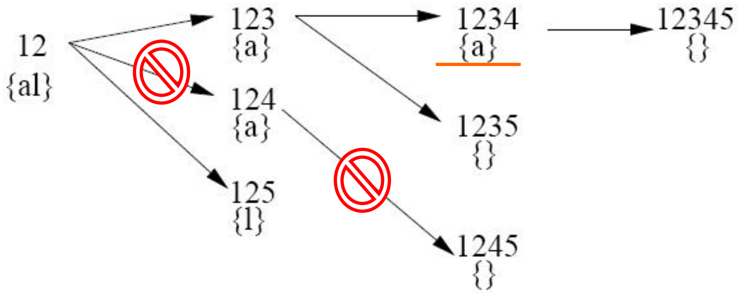
- If rows appear in all tuples of the conditional transposed table, then such branch needs to prune and reconstruct



f_j	$\mathcal{R}(f_j)$
a	4
e	4
h	4

Prune method 3

- In each node, if corresponding support features is found, prune out the branch.

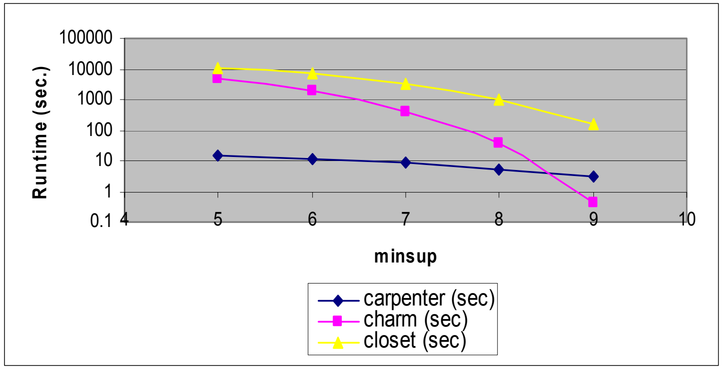


Performance

- CARPENTER is comparing with CHARM and CLOSET
  - Both CHARM and CLOSET use column enumeration approach
- Use lung cancer dataset
  - 181 samples with 12533 features
- Two parameters: minsup and length ratio
  - Length ratio is the percentage of column from original dataset

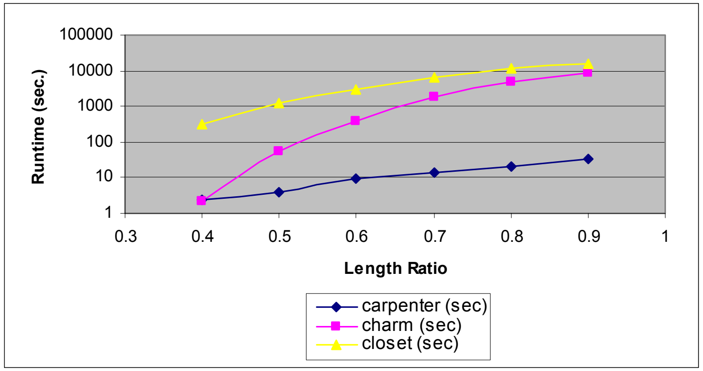
Performance

- Length ratio =60%, varying minsup



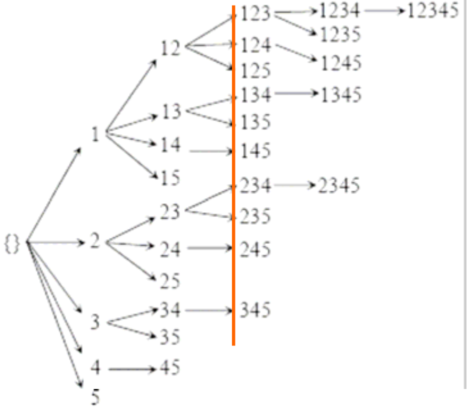
Performance

- Minsup=4% varying length ratio



Comments

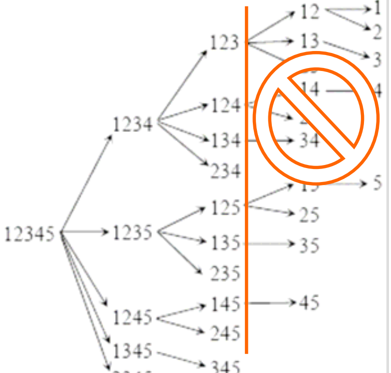
- Bottom-up approach of CARPENTER is not efficient.



minsup=3

Comments

- TD-Close uses top-down approach.



minsup=3

Conclusion

- CARPENTER is used to find the frequent closed pattern in biological dataset.
- CARPENTER uses row enumeration instead of column enumeration to overcome the high dimensionality of biological datasets.
- Not very efficient somehow

References

- A. K. H. Tung J. Yang F. Pan, G. Cong and M. J. Zaki. CARPENTER: Finding closed patterns in long biological datasets. In *In Proc. 2003 ACM SIGKDD Int. Conf. On Knowledge Discovery and Data Mining*, 2003.

Thank you!

Questions?