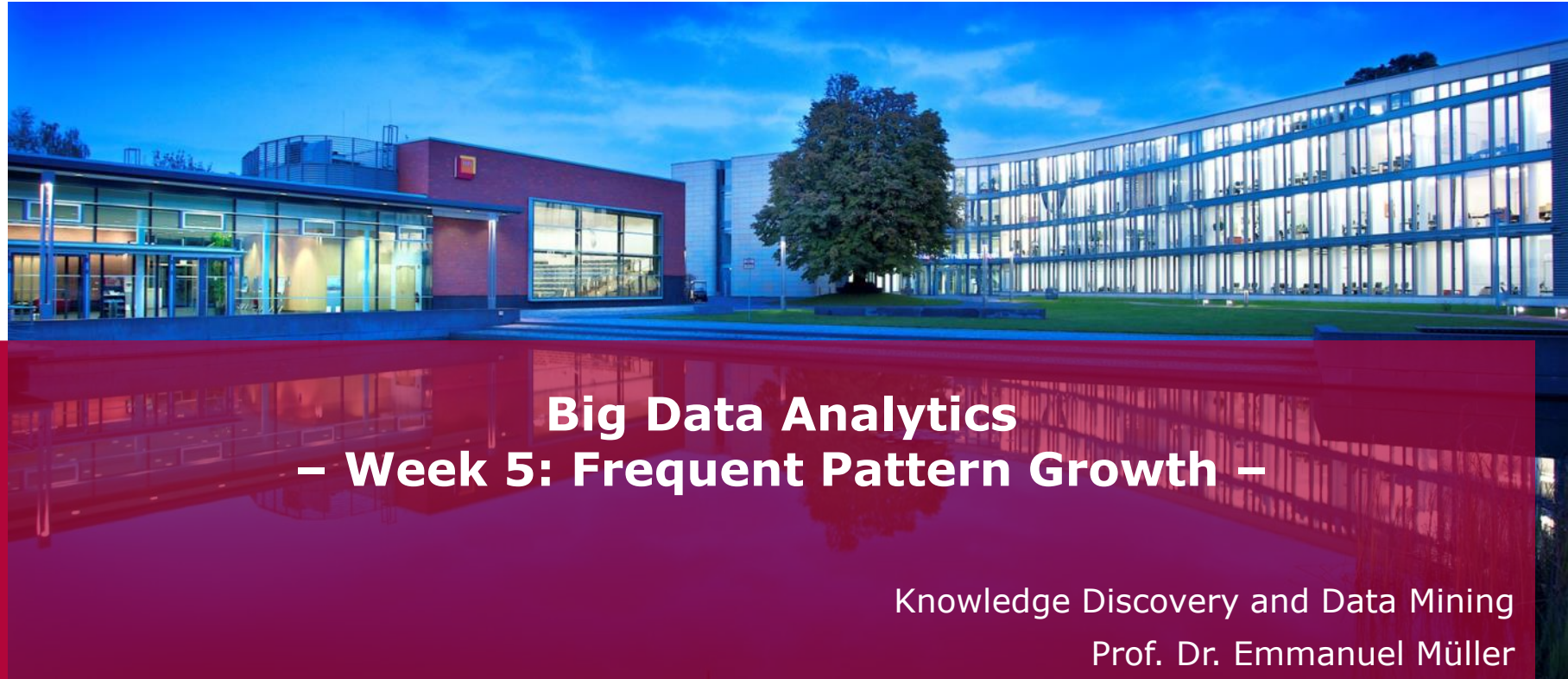




Big Data Analytics

– Week 5: Frequent Pattern Growth –

Knowledge Discovery and Data Mining
Prof. Dr. Emmanuel Müller

Mining Frequent Patterns Using FP-tree

- General idea (divide-and-conquer)
 - Recursively grow frequent pattern path using the FP-tree

- Method
 - For each item, construct its **conditional pattern-base** (*prefix paths*), and then its **conditional FP-tree**
 - Repeat the process on each newly created conditional FP-tree ...
 - ...until the resulting FP-tree is **empty**, or it contains **only one path** (single path will generate all the combinations of its sub-paths, each of which is a frequent pattern)

Major Steps to Mine FP-tree

1. Construct conditional pattern base for each node in the FP-tree
2. Construct conditional FP-tree from each conditional pattern-base
3. Recursively mine conditional FP-trees and grow frequent patterns obtained so far
 - If the conditional FP-tree contains a single path, simply enumerate all the patterns

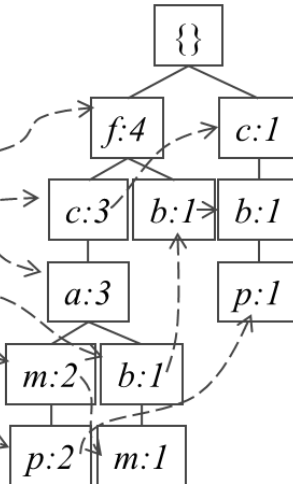
Major Steps to Mine FP-tree: Conditional Pattern Base

1. Construct conditional pattern base for each node in the FP-tree

- Starting at the frequent header table in the FP-tree
- Traverse the FP-tree by following the link of each frequent item
- Accumulate all of transformed prefix paths of that item to form a conditional pattern base
 - For each item its prefixes are regarded as condition for it being a suffix. These prefixes form the conditional pattern base. The frequency of the prefixes can be read in the node of the item.

header table:

item	frequency	head
<i>f</i>	4	•
<i>c</i>	4	•
<i>a</i>	3	•
<i>b</i>	3	•
<i>m</i>	3	•
<i>p</i>	3	•



conditional pattern base:

item	cond. pattern base
<i>f</i>	{}
<i>c</i>	<i>f</i> :3
<i>a</i>	<i>fc</i> :3
<i>b</i>	<i>fca</i> :1, <i>f</i> :1, <i>c</i> :1
<i>m</i>	<i>fca</i> :2, <i>fcab</i> :1
<i>p</i>	<i>fcam</i> :2, <i>cb</i> :1

- Node-link property
 - For any frequent item a_i , all the possible frequent patterns that contain a_i can be obtained by following a_i 's node-links, starting from a_i 's head in the FP-tree header

- Prefix path property
 - To calculate the frequent patterns for a node a_i in a path P , only the prefix sub-path of a_i in P needs to be accumulated, and its frequency count should carry the same count as node a_i .

Major Steps to Mine FP-tree: Conditional FP-tree

2. Construct conditional FP-tree from each conditional pattern-base

- The prefix paths of a suffix represent the conditional basis.
→ They can be regarded as transactions of a database.
- Those prefix paths whose support $\geq \text{minSup}$, induce a conditional FP-tree
- For each pattern-base
 - Accumulate the count for each item in the base
 - Construct the FP-tree for the frequent items of the pattern base

conditional pattern base:

item	cond. pattern base
<i>f</i>	<i>{}</i>
<i>c</i>	<i>f:3</i>
<i>a</i>	<i>fc:3</i>
<i>b</i>	<i>fca:1, f:1, c:1</i>
<i>m</i>	<i>fca:2, fcab:1</i>
<i>p</i>	<i>fcam:2, cb:1</i>

e.g. item *m*

item	frequency
<i>f</i>	3
<i>c</i>	3
<i>a</i>	3
<i>b</i>	1 <i>x</i>

m-conditional FP-tree

{}|*m*
|
f:3
|
c:3
|
a:3

Conditional FP-tree

2. Construct conditional FP-tree from each conditional pattern-base

conditional pattern base:

item	cond. pattern base
<i>f</i>	$\{\}$
<i>c</i>	$f:3$
<i>a</i>	$fc:3$
<i>b</i>	$fca:1, f:1, c:1$
<i>m</i>	$fca:2, fcab:1$
<i>p</i>	$fcam:2, cb:1$

$\{\} | f = \{\}$

$\{\} | c$

$\{\} | a$

$\{\} | b = \{\}$

$\{\} | m$

$\{\} | p$

$f:3$

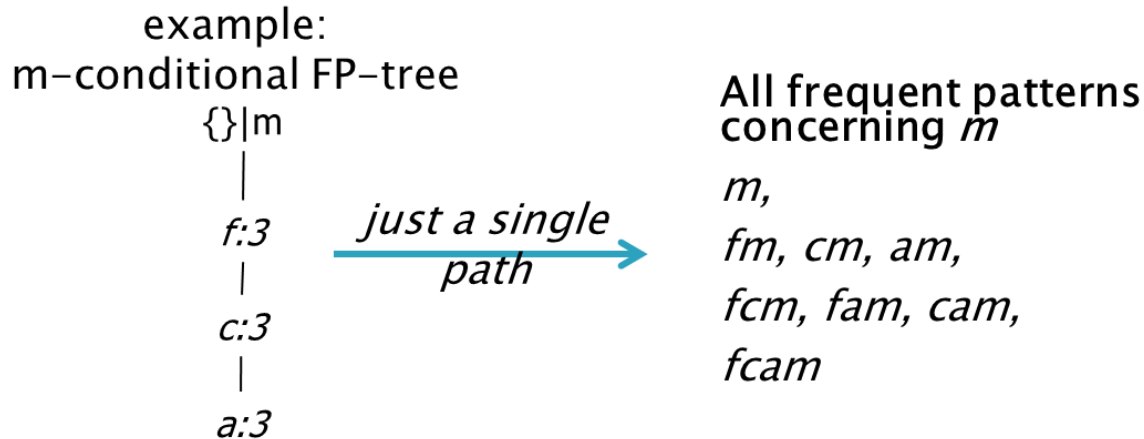
$f:3$
 $c:3$

$f:3$
 $c:3$
 $a:3$

$c:3$

Major Steps to Mine FP-tree

3. Recursively mine conditional FP-trees and grow frequent patterns obtained so far
 - If the conditional FP-tree contains a single path, simply enumerate all the patterns (enumerate all combinations of sub-paths)



FP-tree: Full Example

database:

TID	items bought	(ordered) frequent items
100	{b, c, f}	{f, b, c}
200	{a, b, c}	{b, c}
300	{d, f}	{f}
400	{b, c, e, f}	{f, b, c}
500	{f, g}	{f}

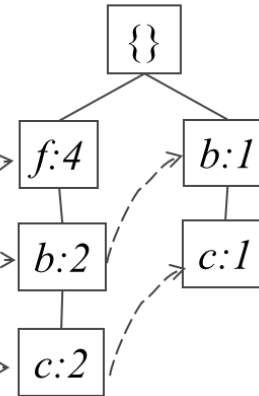
$minSup=0.4$

header table:

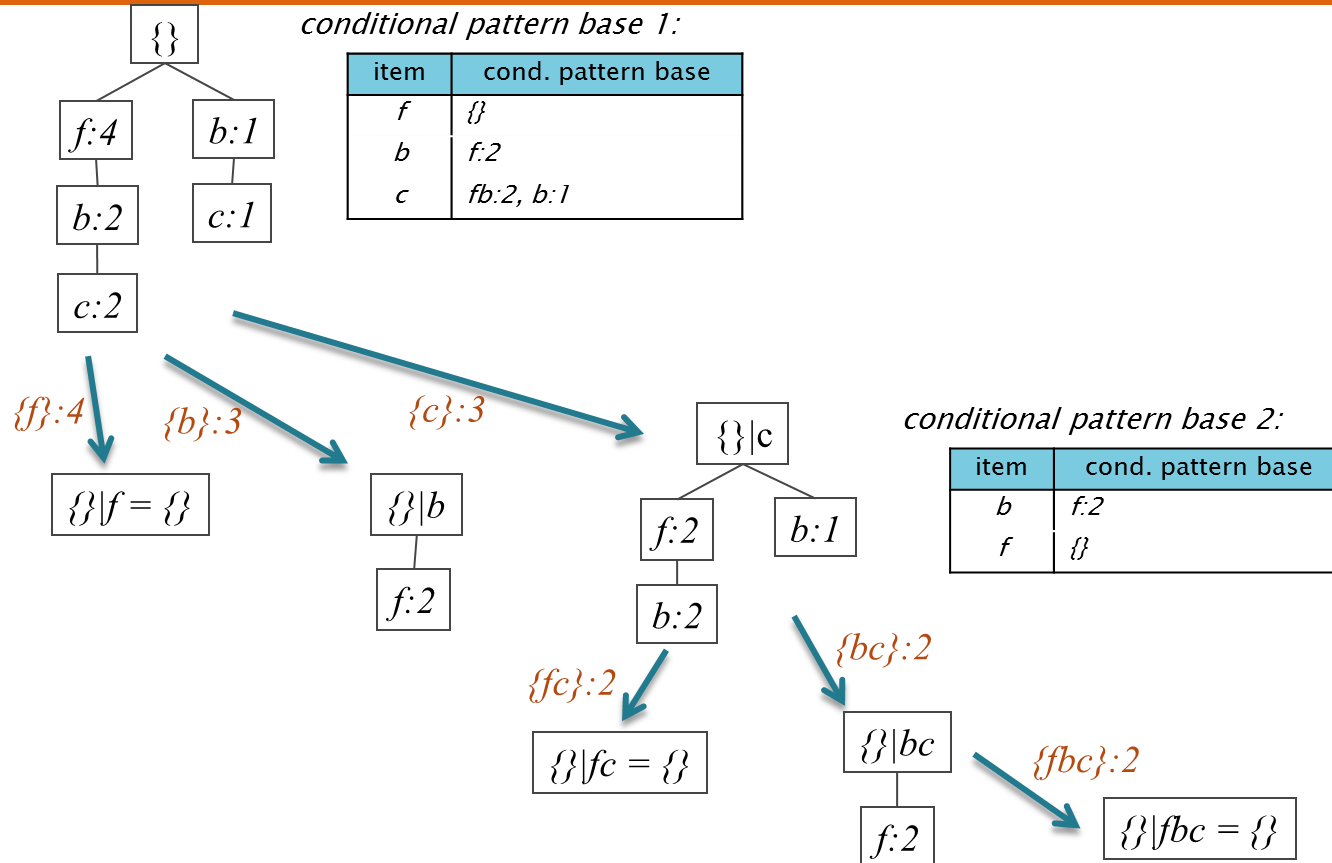
item	frequency	head
f	4	•
b	3	•
c	3	•

conditional pattern base:

item	cond. pattern base
f	{}
b	f:2
c	fb:2, b:1



FP-tree: Full Example



Principles of Frequent Pattern Growth

- Pattern growth property
 - Let α be a frequent itemset in DB, B be α 's conditional pattern base, and β be an itemset in B.

Then $\alpha \cup \beta$ is a frequent itemset in DB iff β is frequent in B.
- "abcdef" is a frequent pattern, if and only if
 - "abcde" is a frequent pattern, and
 - "f" is frequent in the set of transactions containing "abcde"

Why Is Frequent Pattern Growth Fast?

Performance study in [Han, Pei&Yin '00] shows

- FP-growth is an order of magnitude faster than Apriori, and is also faster than tree-projection

Reasoning

- No candidate generation, no candidate test
 - Apriori algorithm has to proceed breadth-first
- Use compact data structure
- Eliminate repeated database scan
- Basic operation is counting and FP-tree building

