1 Association Rule Mining: Basics

1.1 Confidence, Support and Interest

1. $supp(\{5, 9\}) = 5$
2. $supp(\{1, 3, 4, 5\}) = 2$
3. $conf(\{5\} \Rightarrow \{9\}) = \dfrac{5}{supp(\{5\}) = 7} \approx 0.7$
4. $conf(\{3, 4, 5\} \Rightarrow \{1\}) = \dfrac{2}{supp(\{3, 4, 5\}) = 6} = \dfrac{1}{3}$
5. $interest(\{5\} \Rightarrow \{9\}) = conf(5 \Rightarrow 9) - supp(9)/|D| = 5/7 - 8/10 \approx -0.1$
6. $interest(\{3, 4, 5\} \Rightarrow \{1\}) = conf(3, 4, 5 \Rightarrow 1) - supp(1)/|D| = 1/3 - 6/10 \approx -0.27$

1. $\{3, 4, 5, 9\} \Rightarrow \{10\}$
2. $supp(\{3, 4, 5, 9, 10\}) = 5$
3. $conf(\{3, 4, 5, 9\} \Rightarrow \{10\}) = \dfrac{5}{supp(\{3, 4, 5, 9\}) = 5} = 1.0$
4. $\{3\} \Rightarrow \{4, 5, 9, 10\}$
5. $conf(\{3\} \Rightarrow \{4, 5, 9, 10\}) = \dfrac{5}{supp(\{3\}) = 6} = \dfrac{5}{6} \approx 0.83$

1.2 Lift

Situation 1 $lift(ML \Rightarrow DM) = \dfrac{50/300}{200/500} = 0.417$

Situation 2 $lift(H \Rightarrow K) = \dfrac{200/600}{300/1000} = 1.11$

2 Closed and Maximal Itemsets

In the following table, which itemsets are frequent, frequent closed and frequent maximal for the min.support threshold 10.

Itemset	support	frequent	closed	maximal
A	15	x		
B	20	x	x	
C	33	x	x	
D	25	x		
AB	15	x	x	
AC	12	x		
AD	15	x	x	
BC	18	x	x	
BD	5			
CD	25	x	x	
ABC	10	x	x	x
ABD	2			
ACD	12	x	x	x
BCD	3			
ABCD	1			

3 Apriori: Join and Prune

1. Generate all legal candidates of the next level of Apriori's search:
   $\{1, \underline{3, 5, 10}\}$,$\{2, \underline{3, 4, 5}\}$,$\{3, \underline{8, 9, 10}\}$
2. All pruned: Missing subsets underlined above.

4 Sequence mining

For the transactional database below:

1. Convert it to a customer-sequence database.
2. Mine the *a*-projected database and the *c*-projected database using FREESPAN with $minsup = 2$.
3. Perform the same task using PREFIXSPAN.

Sequence database

CID	Sequence
1	(ab)c(ad)bgf
2	(bh)cagf(ac)h(cd)
3	i(bgf)ijc(bg)
4	(bc)dcfa(bc)ad(cd)

FreeSpan

1. Compute frequencies of individual items (*f-list*): $c:4,\ b:4,\ f:4,\ a:3,\ d:3,\ g:3,\ h:1,\ i:1,\ j:1$
2. Remove infrequent items from the database:

Id	Seq
1	(ab)c(ad)bgf
2	bcagf(ac)(cd)
3	(bgf)c(bg)
4	(bc)dcfa(bc)ad(cd)

3. Mining *c-projection* – sequential patterns only containing c:

1'	c
2'	ccc
3'	c
4'	cccc

Generate candidate 2-sequences and count their support: $cc:2$

   (a) *cc-projection* (only transactions that contain cc):

2"	ccc
4"	cccc

   Candidate 3-sequences: $ccc:2$

   (b) *ccc-projection* (only transactions that contain ccc):

2"'	ccc
4"'	cccc

   Candidate 4-sequences: ~~$cccc:1$~~
   There are no frequent candidate sequence $\Rightarrow$ terminate.

4. Mining *a-projection* – sequential patterns only containing a, and c, b, and f:

1'	(ab)cabf
2'	bcaf(ac)c
4'	(bc)cfa(bc)ac

Generate candidate 2-sequences and count their support:
$aa:3,\ ac:3,\ ca:3,$ ~~$(ac):1$~~, $ab:2,\ ba:3,$ ~~$(ab):1$~~, $af:2,\ fa:2,$ ~~$(af):0$~~.

For each frequent 2-sequence, mine its projection, e.g.:

   (a) *ab-projection*:

1"	(ab)cabf
4"	(bc)cfa(bc)ac

   Candidate 3-sequences: $ab\mathbf{a}$, $ab\mathbf{b}$, $ab\mathbf{c}$, $ab\mathbf{f}$, $\mathbf{a}ab$, $\mathbf{b}ab$, $\mathbf{c}ab$, $\mathbf{f}ab$, $\mathbf{a}(\mathbf{ab})\ldots$

(b) *ba-projection* (only transactions that contain ba):

1”	(ab)cabf
2”	bcaf(ac)c
4”	(bc)cfa(bc)ac

Candidate 3-sequences: note that $ba\mathbf{b}$ can be generated again.

(c) *etc.*

The description above corresponds to a *naïve* version of FREESPAN, which essentially attempts a straightforward translation of FP-GROWTH to sequential pattern mining. The description aims at illustrating the shortcomings of this translation.
The actual FREESPAN algorithm uses a number of data structures that help to avoid generating redundant candidates. See the original paper (link available on Toledo). This mitigates certain technical, but not conceptual shortcomings of the naïve version.

PrefixSpan

1. Compute frequencies of individual items: $a:3,\ b:4,\ c:4,\ d:3,\ f:4,\ g:3,\ h:1,\ i:1,\ j:1$
2. Remove infrequent items from the database:

Id	Seq
1	(ab)c(ad)bgf
2	bcagf(ac)(cd)
3	(bgf)c(bg)
4	(bc)dcfa(bc)ad(cd)

3. Mining *c-projection*:

1’	(ad)bgf
2’	agf(ac)(cd)
3’	(bg)
4’	dcfa(bc)ad(cd)

Compute frequencies of individual items (*f-list*): $a:3,\ b:3, c:3,\ d:2,\ f:3,\ g:3$

Recursively mine projections:

(a) ca-projection (i.e. a-projection of c-projection)

1”	(_d)bgf
2”	gf(ac)(cd)
4”	(bc)ad(cd)

(b) cb-projection

1’	gf
3’	(_g)
4’	(_c)ad(cd)

(c) cc-projection…

(d) cd-projection…

(e) cf-projection…

(f) cg-projection…