WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Finding Frequent Itemsets and Association Rules

Marzena Kryszkiewicz

HUMAN CAPITAL
Project is co-financed by European Union within European Social Fund

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Basic Notions and Properties

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Association Rules - Informally

- Let item {*fish*} occur in 5% of sales transactions and set {*fish, white wine*} occur in 4% of them. This information allows us to derive an *association rule* stating that:

4 out of 5 *customers; that is,* 80% of *customers who buy fish also buy white wine*.

- In order to derive such rules we need to know how many transactions support respective *sets of items* (or *itemsets*).

3

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Support of Itemsets

- Let dataset D be a set of *transactions*, where each transaction is a subset of items in *I*.
- *Support of an itemset X*, denoted by *sup*(*X*), is the number of transactions in D that contain all items in *X*; that is,

$$sup(X) = |\{T \in D \mid X \subseteq D\}|.$$

4

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Example: Supports of Itemsets

Example dataset D

Id	*Transaction*
T_1	*ABCDEG*
T_2	*ABCDEF*
T_3	*ABCDEH*
T_4	*ABDE*
T_5	*ACDEH*
T_6	*BCE*

- *sup*(*ABC*) = 3.
- *sup*(*EH*) = 2.
- Supports of all supersets of *EH* are not greater than 2 either.
- Supports of all subsets of *EH* can be greater than 2.

5

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Relative Support of Itemsets

- *Relative support of an itemset X,* denoted by *rSup*(*X*), is the ratio of the number of the transactions in D that contain all items in *X* to the number of all transactions in D:

$$rSup(X) = sup(X) / |D|.$$

- **Remark:** *rSup*(*X*) can be regarded as an estimation of the probability of the occurrence of itemset *X* in D.

6

Example: Relative Supports

Example dataset D

Id	*Transaction*
T_1	*ABCDEG*
T_2	*ABCDEF*
T_3	*ABCDEH*
T_4	*ABDE*
T_5	*ACDEH*
T_6	*BCE*

- *rSup*(*ABC*) = 3/6 = 50%,
- *rSup*(*EH*) = 2/6 ≈ 33%.

7

Frequent Itemsets

- *X* is defined a *frequent itemset* if
  sup(*X*) > *minSup*,
  where *minSup* is the user-defined threshold value.
- **Basic property of itemsets**:Supports of supersets of an itemset *X* are not greater than *sup*(*X*).

8

Example: In(frequent) Itemsets

Example dataset D

Id	*Transaction*
T_1	*ABCDEG*
T_2	*ABCDEF*
T_3	*ABCDEH*
T_4	*ABDE*
T_5	*ACDEH*
T_6	*BCE*

- *sup*(*ABC*) = 3, *sup*(*EH*) = 2.
- Let *minSup* = 2. Then: *ABC* is frequent, *EH* is not frequent.
- Supports of all supersets of *EH* are not greater than 2 either, hence supersets of *EH* are not frequent.
- However, supports of subsets of *EH* can be greater than 2. Thus, it may happen that subsets of *EH* are frequent.

9

Association Rules (ARs)

- An *association rule* is an expression associating two itemsets:
  $$X \rightarrow Y,$$
  where $\emptyset \neq Y \subseteq I$ and $X \subseteq I \setminus Y$.
- *X* is called an *antecedent* of $X \rightarrow Y$.
- *Y* is called a *consequent* of $X \rightarrow Y$.
- $X \rightarrow Y$ is said to be *based on* $X \cup Y$, and $X \cup Y$ is called the *base* of $X \rightarrow Y$.

10

Support of Association Rule

- *Support* of $X \rightarrow Y$ is defined as the number of transactions that contains the base of $X \rightarrow Y$; that is,
  $sup(X \rightarrow Y) = sup(X \cup Y)$.
- *Relative support* of $X \rightarrow Y$ is defined as the relative support of its base:
  $rSup(X \rightarrow Y) = rSup(X \cup Y)$.

11

Confidence of Association Rule

- *Confidence* of $X \rightarrow Y$ is defined as the ratio of the number of transactions that contain the base $X \cup Y$ to the number of transactions containing the antecedent *X*:
  $conf(X \rightarrow Y) = sup(X \rightarrow Y) / sup(X)$.
- **Remark:** *conf*($X \rightarrow Y$) can be regarded as an estimation of the conditional probability that *Y* occurs in a transaction *T* provided *X* occurs in *T*.

12

Example: Association Rules

Example dataset D

Id	Transaction
T_1	ABCDEG
T_2	ABCDEF
T_3	ABCDEH
T_4	ABDE
T_5	ACDEH
T_6	BCE

$sup(ABC) = 3$, $sup(A) = 5$.

Hence:

- $sup(\{A\} \rightarrow \{BC\}) = sup(\{ABC\}) = 3$,
- $conf(\{A\} \rightarrow \{BC\}) = sup(\{ABC\}) / sup(\{A\}) = 3/5$.

13

Strong Association Rules

- *Strong association rules* (***AR***) are defined as those rules in *AR* whose support is above *minSup* and confidence is above *minConf*; that is,

$$\boldsymbol{AR} = \{X \rightarrow Y \in AR \mid sup(X \rightarrow Y) > minSup \wedge conf(X \rightarrow Y) > minConf\},$$

where $minSup \in [0, |D|)$ and $minConf \in [0, 1)$.

14

Strong ARs and Frequent Itemsets

$$\boldsymbol{AR} = \{X \rightarrow Y \in AR \mid \boldsymbol{sup(X \rightarrow Y) > minSup} \wedge conf(X \rightarrow Y) > minConf\}$$

$$= \{X \rightarrow Y \in AR \mid \boldsymbol{sup(X \cup Y) > minSup} \wedge conf(X \rightarrow Y) > minConf\}$$

$$= \{X \rightarrow Y \in AR \mid \boldsymbol{(X \cup Y) \text{ is frequent}} \wedge conf(X \rightarrow Y) > minConf\}$$

15

Discovery of Strong Association Rules

AR is discovered in two steps:

- Find frequent itemsets ***F*** and their supports in dataset D.
- Generate ***AR*** only from ***F***: Let $Z \in \boldsymbol{F}$, $Z \neq \emptyset$ and $Y \subseteq Z$. Then, any candidate rule $Z \setminus Y \rightarrow Y$ is a strong association one if:

$$sup(Z) / sup(Z \setminus Y) > minConf.$$

16

Finding Frequent Itemsets and Association Rules with Apriori

Finding Frequent Itemsets

- Within each iteration *i*:
  - Determine supports of candidate itemsets of length *i*.
  - From those candidates of length *i* that turned out frequent, create candidates of length *i* + 1.

18

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Frequent 1-Itemsets

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

- Let minSup = 1
- Iteration 1:
  $\mathbf{C_0 \rightarrow F_0}$: $\emptyset_8$
  $\mathbf{C_1 \rightarrow F_1}$: $a_6\ b_5\ c_4\ e_4\ f_4\ h_3$

19

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Frequent 2-Itemsets

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

- Let minSup = 1
- After iteration 1:
  $\mathbf{F_1}$: $a_6\ b_5\ c_4\ e_4\ f_4\ h_3$
- Iteration 2:
  $\mathbf{C_2 \rightarrow F_2}$: $ab_4\ ac_4\ ae_3\ af_3\ ah_2\ bc_3\ be_4\ bf_2\ bh_1$
  $ce_2\ cf_2\ ch_2\ ef_2\ eh_0\ fh_1$

Itemsets found as infrequent after support calculation.

20

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Frequent 3-Itemsets

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

- Let minSup = 1
- After iteration 2:
  $\mathbf{F_2}$: $ab_4\ ac_4\ ae_3\ af_3\ ah_2\ bc_3\ be_4\ bf_2$
  $ce_2\ cf_2\ ch_2\ ef_2$
- Iteration 3:
  $\mathbf{C_3 \rightarrow F_3}$: $abc_3\ abe_3\ abf_1\ abh\ ace_2\ acf_2\ ach_2$
  $aef_1\ aeh\ afh\ bce_2\ bcf_1\ bef_2\ cef_1\ ceh$
  cfh

Itemsets found as infrequent as supersets of infrequent itemsets.

21

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Frequent 4-Itemsets

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

- Let minSup = 1
- After iteration 3:
  $\mathbf{F_3}$: $abc_3\ abe_3\ ace_2\ acf_2\ ach_2\ bce_2\ bef_2$
- Iteration 4:
  $\mathbf{C_4 \rightarrow F_4}$: $abce_2\ acef\ aceh\ acfh$

22

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Frequent 5-Itemsets

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

- Let minSup = 1
- After iteration 4:
  $\mathbf{F_4}$: $abce_2$
- Iteration 5:
  $\mathbf{C_5}$: -

23

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Found Frequent Itemsets

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

$\emptyset_8$
$a_6\ b_5\ c_4\ e_4\ f_4\ h_3$
$ab_4\ ac_4\ ae_3\ af_3\ ah_2\ bc_3\ be_4\ bf_2\ ce_2\ cf_2\ ch_2\ ef_2$
$abc_3\ abe_3\ ace_2\ acf_2\ ach_2\ bce_2\ bef_2$
$abce_2$

- **Note:** Let *n* be the length of a longest frequent itemset.
  Apriori finds in either *n* or *n*+1 iterations all frequent itemsets.

24

Properties of the Apriori method of creating candidates…

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

- Let minSup = 1
- After iteration 3:
  $\mathbf{F_3}$: abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2
- Iteration 4:
  $\mathbf{C_4 \to F_4}$: $abce_2$ *acef aceh acfh*
- Note: *abc* and *abe* are parents of *abce*.
- **General observation.** *Parents of each candidate of length i* are *its first two frequent subsets of length i – 1.*

25

Properties of the Apriori method of creating candidates…

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

- Let minSup = 1
- After iteration 3:
  $\mathbf{F_3}$: abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2
- Iteration 4:
  $\mathbf{C_4 \to F_4}$: $abce_2$ *acef aceh acfh*
- Note: *ace* and *acf* are parents of *acef*.

26

Properties of the Apriori method of creating candidates

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

- Let minSup = 1
- After iteration 3:
  $\mathbf{F_3}$: abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2
- Iteration 4:
  $\mathbf{C_4 \to F_4}$: $abce_2$ *acef aceh acfh*
- Why was itemset *bcef* not created?
  Because its second parent *bcf* $\notin$ $\mathbf{F_3}$.

27

Discovery of Association Rules with AprioriRuleGen…

- Candidate rules are built from each non-empty frequent itemset.
- Let *Z* be a given non-empty frequent itemset. In iteration *i*, candidate rules of the form:

$$Z \setminus Y \to Y,$$

where $Y \subset Z$ and $|Y| = i$.

28

Discovery of Association Rules with AprioriRuleGen…

- In iteration *i*+1, consequents of candidates rules of base *Z* are created by merging *i*-item consequents of strong association rules of base *Z* that were found in the previous iteration.
- **Property.** Let r_1: $Z \setminus Y \to Y$ and r_2: $Z \setminus Y' \to Y'$, where $Y \subset Y'$, be association rules.
  - $conf(r_1) \geq conf(r_2)$,
  - If $conf(r_1) \leq minConf$, then $conf(r_2) \leq minConf$.

29

Example of Useless Creation of Candidate Rules

- **Example.** Let $minSup = 1$, $minConf = 60\%$, base = $\{abce\}_2$, r_1: $ce \to ab$ [2, 2/2] and r_2: $be \to ac$ [2, 2/4] be candidate rules considered in iteration 2. So, r_1 is strong, while r_2 is not strong.

  Let us consider rule r_3 of base {*abce*} whose consequent is the union of the consequents of rules r_1 and r_2; that is, rule r_3: $e \to abc$. Hence:

  – $sup(r_3) = sup(\{abce\}) = 2$,

  – $conf(r_3) = \frac{sup(\{abce\})}{sup(\{e\})} \leq \frac{sup(\{abce\})}{sup(\{be\})} = conf(r_2) = \frac{2}{4} < minConf$.

  Thus, r_3 is not strong.

30

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Discovery of **AR**s…

Frequent itemsets (*minSup* = 1): $\emptyset_8$

a_6 b_5 c_4 e_4 f_4 h_3

ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2

abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2

$abce_2$

Let *minConf* = 60%, *Z* = *abce*.

Iteration 1:

- Consequents of candidate rules: **Y**$_1$ = {*a*, *b*, *c*, *e*}.
- Candidate rules:
  - *bce*→*a* [2, 2/2];
  - *ace*→*b* [2, 2/2];
  - *abe*→*c* [2, 2/3];
  - *abc*→*e* [2, 2/3].
- Strong association rules:
  - *bce*→*a* [2, 2/2];
  - *ace*→*b* [2, 2/2];
  - *abe*→*c* [2, 2/3];
  - *abc*→*e* [2, 2/3].

31

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Discovery of **AR**s…

Frequent itemsets: $\emptyset_8$

a_6 b_5 c_4 e_4 f_4 h_3

ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2

abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2

$abce_2$

Iteration 2 (*minConf* = 60%, *Z* = *abce***):**

- Consequents of **AR**s found in iteration 1: **Y**$_1$ = {*a*, *b*, *c*, *e*}.
- Consequents of candidate rules: **Y**$_2$ = {*ab*, *ac*, *ae*, *bc*, *be*, *ce*}.
- Candidate rules:
  - *ce*→*ab* [2, 2/2]; *ae*→*bc* [2, 2/3];
  - *be*→*ac* [2, 2/4]; *ac*→*be* [2, 2/4];
  - *bc*→*ae* [2, 2/3]; *ab*→*ce* [2, 2/4];
- Strong association rules:
  - *ce*→*ab* [2, 2/2];
  - *bc*→*ae* [2, 2/3];
  - *ae*→*bc* [2, 2/3].

32

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Discovery of **AR**s…

Frequent itemsets (*minSup* = 1): $\emptyset_8$

a_6 b_5 c_4 e_4 f_4 h_3

ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2

abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2

$abce_2$

Iteration 3 (*minConf* = 60%, *Z* = *abce***):**

- Consequents of **AR**s found in iteration 2: **Y**$_2$ = {*ab*, *ae*, *bc*}.
- Consequents of candidate rules: **Y**$_3$ = {*abe*}.
- Candidate rules:
  - *c*→*abe* [2, 2/4];
- Strong association rules:
  - *None*

33

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Found **AR**s

Frequent itemsets (*minSup* = 1): $\emptyset_8$

a_6 b_5 c_4 e_4 f_4 h_3

ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2

abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2

$abce_2$

Strong association rules (*minConf* = 60%, *Z* = *abce*):

- *bce*→*a* [2, 2/2];
- *ace*→*b* [2, 2/2];
- *abe*→*c* [2, 2/3];
- *abc*→*e* [2, 2/3];
- *ce*→*ab* [2, 2/2];
- *bc*→*ae* [2, 2/3];
- *ae*→*bc* [2, 2/3].

34

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Important Operations in Apriori and AprioriRuleGen

- An important time-consuming operation in *Apriori* is searching *i* item candidates supported by a given transaction.
- An important time-consuming operation in *AprioriRuleGen* is searching frequent *i* itemsets (candidate rule consequents) of a given frequent itemset in order to learn their supports.
- Thus, in both cases *i* item subsets of a given itemset are searched.

35

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Usage of a Hash Tree

- A hash tree is used in order to make the identification of *i* item subsets of a given itemset efficient.
- In particular, all *i* item candidate sets are stored in a hash tree.

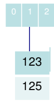
36

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Creation of a Hash Tree with 3-Itemsets Candidates…

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets



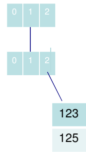
37

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Creation of a Hash Tree with 3-Itemsets Candidates…

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets



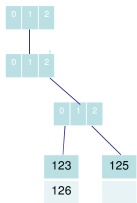
38

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Creation of a Hash Tree with 3-Itemsets Candidates…

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets



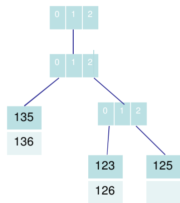
39

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Creation of a Hash Tree with 3-Itemsets Candidates…

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets



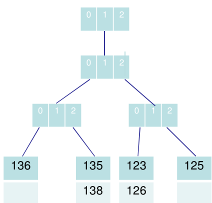
40

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Creation of a Hash Tree with 3-Itemsets Candidates…

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets



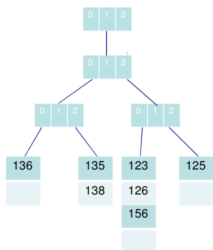
41

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Creation of a Hash Tree with 3-Itemsets Candidates…

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets

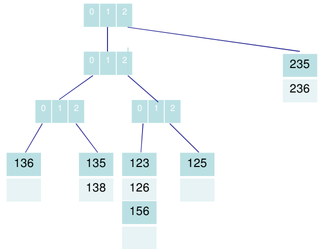


42

Example: Creation of a Hash Tree with 3-Itemsets Candidates...

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets

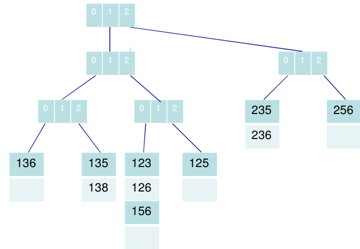


43

Example: Creation of a Hash Tree with 3-Itemsets Candidates…

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets

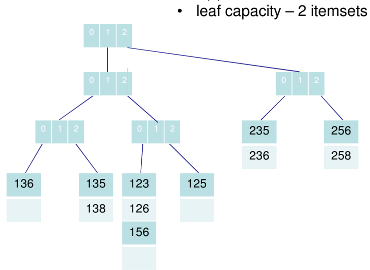


44

Example: Creation of a Hash Tree with 3-Itemsets Candidates

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets

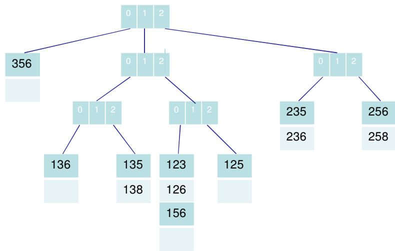


45

Example: Candidate 3-Itemsets in a Hash Tree

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Assumptions:
- h(*x*) = *x* mod 3
- leaf capacity – 2 itemsets



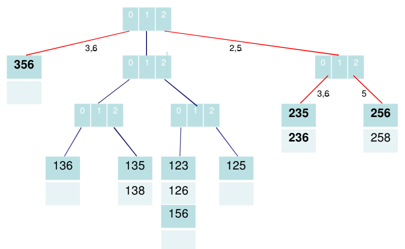
46

Example: Searching for Subsets in a Hash Tree…

candidate	coded candidate
abc	123
abe	125
abf	126
…	…

Assumption:
- h(*x*) = *x* mod 3

transaction	coded transaction
…	…
bcef	2356
…	…



- 4 subsets of transaction *bcef* (after coding: 2356) were found.

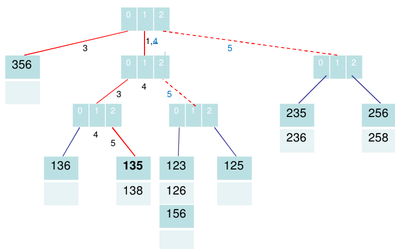
47

Example: Searching for Subsets in a Hash Tree

candidate	coded candidate
abc	123
abe	125
abf	126
…	…

Assumption:
- h(*x*) = *x* mod 3

transaction	coded transaction
…	…
acde	1345
…	…



1 subset of transaction *acde* (after coding: 1345) was found.

48

WARSAW UNIVERSITY OF TECHNOLOGY
DEVELOPMENT PROGRAMME

Finding Frequent Itemsets with Eclat, dEclat & Partition

Calculating *FI*s with Eclat…

Id	Transaction
1	{*abc*}
2	{*abc*}
3	{*abc*}
4	{*ab*}
5	{*bcd*}

- $t(X \cup Y) = t(X) \cap t(Y)$
- $sup(X \cup Y) = |t(X \cup Y)|$

- *Tidlists* (*lists of transaction identifiers*) of length 1:
  - $t(\{a\}) = \{1,2,3,4\}$, so $sup(\{a\}) = 4$
  - $t(\{b\}) = \{1,2,3,4,5\}$, so $sup(\{b\}) = 5$
  - $t(\{c\}) = \{1,2,3,5\}$, so $sup(\{c\}) = 4$
  - …
- Tidlists of length 2:
  - $t(\{ab\}) = t(\{a\}) \cap t(\{b\}) = \{1,2,3,4\} \cap \{1,2,3,4,5\} = \{1,2,3,4\}$, so $sup(\{ab\}) = 4$
  - $t(\{ac\}) = t(\{a\}) \cap t(\{c\}) = \{1,2,3,4\} \cap \{1,2,3,5\} = \{1,2,3\}$, so $sup(\{ac\}) = 3$
  - …

50

Calculating *FI*s with Eclat

Id	Transaction
1	{*abc*}
2	{*abc*}
3	{*abc*}
4	{*ab*}
5	{*bc*}

- $t(X \cup Y) = t(X) \cap t(Y)$
- $sup(X \cup Y) = |t(X \cup Y)|$

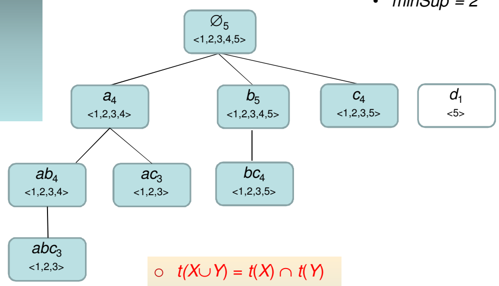
- Tidlists of length 2:
  - $t(\{ab\}) = \{1,2,3,4\}$, $sup(\{ab\}) = 4$
  - $t(\{ac\}) = \{1,2,3\}$, $sup(\{ac\}) = 3$
  - …
- Tidlists of length 3:
  - $t(\{abc\}) = t(\{ab\}) \cap t(\{ac\}) = \{1,2,3,4\} \cap \{1,2,3\} = \{1,2,3\}$, so $sup(\{abc\}) = 3$
  - …

51

Eclat: Calculating *FI*s with Eclat

Id	Transaction
1	{*abc*}
2	{*abc*}
3	{*abc*}
4	{*ab*}
5	{*bcd*}

Assumption:
- *minSup = 2*



- $t(X \cup Y) = t(X) \cap t(Y)$
- $sup(X \cup Y) = |t(X \cup Y)|$

52

Calculating *FI*s with dEclat…

Id	Transaction
1	{*abc*}
2	{*abc*}
3	{*abc*}
4	{*ab*}
5	{*bcd*}

- $d(X \cup Y) = d(Y) \setminus d(X)$
- $sup(X \cup Y) = sup(X) - |d(X \cup Y)|$

- *Difflists* (*differential lists of transaction identifiers*) of length 1:
  - $d(\{a\}) = \{5\}$, so $sup(\{a\}) = sup(\emptyset) - |\{5\}| = 4$
  - $d(\{b\}) = \{\}$, so $sup(\{b\}) = sup(\emptyset) - |\{\}| = 5$
  - $d(\{c\}) = \{4\}$, so $sup(\{c\}) = sup(\emptyset) - |\{4\}| = 4$
  - $d(\{d\}) = \{1,2,3,4\}$, so $sup(\{d\}) = sup(\emptyset) - |\{1,2,3,4\}| = 1$
- *Difflists* of length 2:
  - $d(\{ab\}) = d(\{b\}) \setminus d(\{a\}) = \{\} \setminus \{5\} = \{\}$, so $sup(\{ab\}) = sup(\{a\}) - |\{\}| = 4 - 0 = 4$
  - $d(\{ac\}) = d(\{c\}) \setminus d(\{a\}) = \{4\} \setminus \{5\} = \{4\}$, so $sup(\{ac\}) = sup(\{a\}) - |\{4\}| = 4 - 1 = 3$
  - $d(\{cd\}) = d(\{d\}) \setminus d(\{c\}) = \{1,2,3,4\} \setminus \{4\} = \{1,2,3\}$, so $sup(\{cd\}) = sup(\{c\}) - |\{1,2,3\}| = 4 - 3 = 1$
  - …

53

Calculating *FI*s with dEclat

Id	Transaction
1	{*abce*}
2	{*abc*}
3	{*abc*}
4	{*ab*}
5	{*bcd*}

- $d(X \cup Y) = d(Y) \setminus d(X)$
- $sup(X \cup Y) = sup(X) - |d(X \cup Y)|$

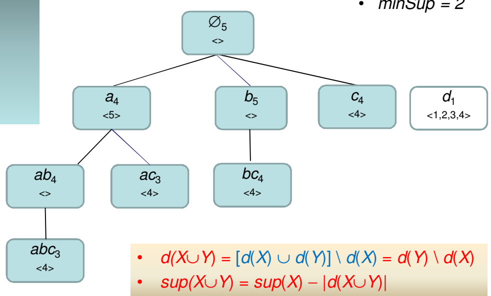
- Difflists of length 2:
  - $d(\{ab\}) = \{\}$, $sup(\{ab\}) = 4$
  - $d(\{ac\}) = \{4\}$, $sup(\{ac\}) = 3$
  - …
- Difflists of length 3:
  - $d(\{abc\}) = d(\{ac\}) \setminus d(\{ab\}) = \{4\} \setminus \{\} = \{4\}$, so $sup(\{abc\}) = sup(\{ab\}) - |\{4\}| = 4 - 1 = 3$
  - …

54

Example: Calculating *FI*s with dEclat

Id	*Transaction*
1	{*abc*}
2	{*abc*}
3	{*abc*}
4	{*ab*}
5	{*bcd*}

Assumption:
- *minSup* = 2



- $d(X \cup Y) = [d(X) \cup d(Y)] \setminus d(X) = d(Y) \setminus d(X)$
- $sup(X \cup Y) = sup(X) - |d(X \cup Y)|$

55

Calculating *FI*s with Partition

Id	*Transaction*	*Block*
1	{*a*}	I
2	{*ab*}	I
3	{*bc*}	I
4	{*acd*}	II
5	{*ac*}	II

- Assumptions:
  - *minSup* = 3
  - Partition of a transaction dataset into *k* = 2 blocks
- Local support threshold
  - $minSup' = \left\lfloor \frac{minSup}{k} \right\rfloor = \left\lfloor \frac{3}{2} \right\rfloor = 1$
- Local frequent itemsets (w.r.t. *minSup*'):
  - $F^{I} = \{a_2, b_2\}$
  - $F^{II} = \{a_2, c_2, ac_2\}$
- (Global) frequent itemsets (w.r.t. *minSup*):
  - $C = F^{I} \cup F^{II} = \{a, b, c, ac\}$
  - $F = \{a_4\}$

○ $t(x_1, \ldots, x_m) = t(x_1) \cap \ldots \cap t(x_m)$

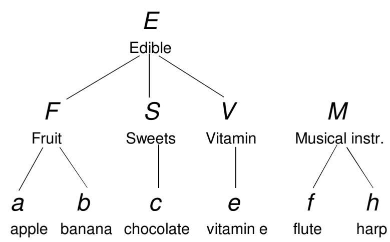
$t(\{abc\}) = t\{a\} \cap t\{b\} \cap t\{c\}$

56

Dealing with Non-Transactional Data

Transactional Data Set + Taxonomies

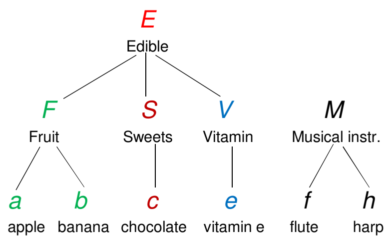
Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*



58

Transactional Data Set with Taxonomies Included in Transactions

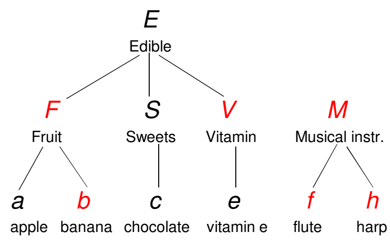
Tid	Items
1	*abceFSVE*
2	*abcefFSVME*
3	*abchFSME*
4	*abeFVE*
5	*acfhFSME*
6	*befFVME*
7	*hM*
8	*afFME*



59

Transactional Data Set + Taxonomies + User Constraints

Tid	Items
1	*~~a~~b~~ce~~FV*
2	*~~a~~b~~ce~~fFMV*
3	*~~a~~b~~c~~hFM*
4	*~~a~~b~~e~~V*
5	*~~ac~~fhFM*
6	*b~~e~~fFMV*
7	*hM*
8	*~~a~~fFM*



60

Transactional Data Set + Negated Items

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

6 positive items {*abcefh*}

6 negated items {*abcefh*}

Tid	Items
1	*abcefh*
2	*abcefh*
3	*abchef*
4	*abecfh*
5	*acfhbe*
6	*befach*
7	*habcef*
8	*afbceh*

Transactional Data Set + Negated Items

Tid	Items
1	*abce*
2	*abcef*
3	*abch*
4	*abe*
5	*acfh*
6	*bef*
7	*h*
8	*af*

6 positive items {*abcefh*}

6 negated items {*abcefh*}

sup({*befh*}) = 2

conf({*bf*}→{*eh*}) = 2/3

conf({*bh*}→{*ef*}) = 2/4

Tid	Items
1	*abcefh*
2	*abcefh*
3	*abchef*
4	*abecfh*
5	*acfhbe*
6	*befach*
7	*habcef*
8	*afbceh*

Transactional Data Set + Negated Items

Tid	Items
1	*abcefh*
2	*abcefh*
3	*abchef*
4	*abecfh*
5	*acfhbe*
6	*befach*
7	*habcef*
8	*afbceh*

Tid	Items
1	1 2 3 4 11 12
2	1 2 3 4 5 12
3	1 2 3 6 11 12
4	1 2 4 10 11 12
5	1 3 5 6 8 10
6	2 4 5 7 9 12
7	6 7 8 9 10 11
8	1 5 8 9 10 12

Item	*a*	*b*	*c*	*e*	*f*	*h*	*a*	*b*	*c*	*e*	*f*	*h*
item id	1	2	3	4	5	6	7	8	9	10	11	12

Transactional Data Set + Negated Items…

Tid	Items
1	*abcefh*
2	*abcefh*
3	*abchef*
4	*abecfh*
5	*acfhbe*
6	*befach*
7	*habcef*
8	*afbceh*

Let n be the number of all distinct items.

Max. number of potentially frequent itemsets without negation = 2^n.

Max. number of potentially frequent itemsets admitting negation = 3^n.

n	max. # itemsets without negation	max. # itemsets admitting negation	difference in orders of magnitude
6	64	729	1
10	1024	59049	2
50	1.13E+15	7.17898E+23	8
100	1.27E+30	5.15378E+47	17
500	3.3E+150	3.636E+238	88

Relational Data → Transactional Data

Height	Colour	Grade
tall	green	5
short	black	4
short	green	4

Tid	Items
1	{1, 3, 6}
2	{2, 4, 5}
3	{2, 3, 5}

Item	(H=tall)	(H=short)	(C=green)	(C=black)	(G=4)	(G=5)
item id	1	2	3	4	5	6
attribute	1	1	2	2	3	3

Relational Data → Transactional Data

Height	Colour	Grade
tall	green	5
short	black	4
short	green	4

Tid	Items
1	{1, 3, 6}
2	{2, 4, 5}
3	{2, 3, 5}

Item	(H=tall)	(H=short)	(C=green)	(C=black)	(G=4)	(G=5)
item id	1	2	3	4	5	6
attribute	1	1	2	2	3	3

{2} → {5} [2, 2/2]. So, (H=short) → (G=4) [2, 100%].

{3} → {5} [1, 1/2]. So, (C=green) → (G=4) [1, 50%].

{2,3} → {5} [1, 1/1]. So, (H=short)∧(C=green) → (G=4) [1, 100%].

References…

– Agrawal R., Imielinski T., Swami A.: Mining Associations Rules between Sets of Items in Large Databases. In: Proc. of the ACM SIGMOD Conference on Management of Data, Washington, USA (1993) 207–216

– Rakesh Agrawal, Ramakrishnan Srikant: Fast Algorithms for Mining Association Rules in Large Databases. VLDB 1994: 487-499

– Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A.I.: Fast Discovery of Association Rules. In: Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI, CA (1996) 307–328

67

References

– Jiawei Han, Micheline Kamber, Jian Pei: Data Mining: Concept and Techniques, The Morgan Kaufmann Series in Data Management Systems, 2011

– Kryszkiewicz, M.: Concise Representations of Frequent Patterns and Association Rules. Prace Naukowe Politechniki Warszawskiej. Elektronika 142. Publishing House of the Warsaw University of Technology (2002)

– Ashok Savasere, Edward Omiecinski, Shamkant B. Navathe: An Efficient Algorithm for Mining Association Rules in Large Databases. VLDB 1995: 432-444

– Mohammed Javeed Zaki, Karam Gouda: Fast vertical mining using diffsets. KDD 2003: 326-335

68

Additional References – Other Algorithms for Discovery of Frequent Itemsets

– Ferenc Bodon: A fast APRIORI implementation. FIMI 2003

– Chen Wang, Mingsheng Hong, Jian Pei, Haofeng Zhou, Wei Wang, Baile Shi: Efficient Pattern-Growth Methods for Frequent Tree Pattern Mining. PAKDD 2004: 441-451

– Frequent Itemset Mining Implementations Repository: http://fimi.ua.ac.be/src/

69