Representing and Discovering Frequent Patterns with Negation

Marzena Kryszkiewicz
Institute of Computer Science
Warsaw University of Technology

1

Layout

- Quick reminding of basic notions
- Deriving supports of patterns with negation
- Deriving supports of patterns by means of generalized disjunctive rules
- Representations of frequent patterns using generalized disjunctive rules
- Naive approach to computing the GDFLR representation
- Upper bound on the length of elements in the GDFLR representation
- Advanced approach to computing the GDFLR representation
- Some experimental results
- Summary

2

Informal introduction to problem

- Let item {*fish*} occur in 5% of sales transactions and set {*fish, white wine*} occur in 4% of transactions.
This information allows us to derive an *association rule* stating that 80% of *customers who buy fish also buy white wine*.
- Let item {coke} occur in 3% of sales transactions and set {*coke, chips,-beer, -milk*} occur in 4% of transactions.
A sample association rule with negation could state that 75% of *customers who buy coke also buy chips and neither beer nor milk*.
- In order to derive such rules we need to know how many transactions support respective *sets of items* (or *itemsets*).

3

Quick reminding of basic notions

- Let $I = \{i_1, i_2, ..., i_m\}$ be a set of distinct *items*.
- Let dataset D be a set of *transactions* (or *records*), where each transaction is a subset of I.
- *Support of an itemset* (or *pattern*) X, denoted by $sup(X)$, is the number of transactions in D that contain all items in X.
- An itemset X is defined *frequent*, 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 $sup(X)$.

4

Quick reminding of basic notions

- A pattern consisting of items $x_1, \ldots, x_m$ and negations of items $x_{m+1}, \ldots, x_n$ will be denoted by $\{x_1, \ldots, x_m, -x_{m+1}, \ldots, -x_n\}$.
- *Support of pattern* $\{x_1, \ldots, x_m, -x_{m+1}, \ldots, -x_n\}$ is the number of transactions in which all items in set $\{x_1, \ldots, x_m\}$ occur and no item in set $\{x_{m+1}, \ldots, x_n\}$ occurs.
- A pattern X is called *positive*, if it does not contain any negated item. Otherwise, X is called a *pattern with negation*.
- A pattern obtained from X by negating any number of items in X is called a *variation of X* .

5

Quick reminding of basic notions

Sample database D.

Id	*Transaction*
T_1	*acefh*
T_2	*af*
T_3	*abch*
T_4	*abe*
T_5	*abce*
T_6	*abcef*
T_7	*bef*
T_8	*h*

$sup(\{a(-b)\}) = 2$

$\{ab\}$ is a *positive pattern*
$\{a(-b)\}$ is a *pattern with negation*

All variations of $\{ab\}$:
$\{ab\}_{[4]}, \{a(-b)\}_{[2]}, \{(-a)b\}_{[1]}, \{(-a)(-b)\}_{[1]}$

6

Calculating supports of patterns with negation

$sup(X(-a)) = sup(X) - sup(Xa)$

$sup(X(-a_1)\ldots(-a_n)) = \Sigma_{Z \subseteq \{a_1, \ldots, a_n\}} (-1)^{|Z|} \times sup(XZ)$

Id	*Transaction*
T_1	***abc**e*
T_2	***abc**h*
T_3	***abc**ef*
T_4	*abe*
T_5	*acefh*
T_6	*bef*
T_7	*h*
T_8	*af*

$sup(abc(-f)) = sup(abc) - sup(abcf) = 3-1 = 2$

$sup(abe(-f)(-h)) = sup(abe(-f)) - sup(abe(-f)h) =$
$(sup(abe) - sup(abef)) - (sup(abeh) - sup(abefh)) =$
$sup(abe) - sup(abef) - sup(abeh) + sup(abefh) =$
$3-1-0+0=2$

7

Pattern with negation may be more frequent

Id	*Transaction*
T_1	*abce*
T_2	*h*
T_3	*abch*
T_4	*abe*
T_5	*acefh*
T_6	*bef*
T_7	*abcef*
T_8	*af*

$sup(\{f\}) = 4$; $sup(\{fh\}) = 1$; $sup(\{f(-h)\}) = 3$.

Let $minSup = 2$. Then:

$\{f\}$ is frequent,

$\{fh\}$ is infrequent,

$f(-h)\}$ is frequent.

$sup(\{f(-h)\}) = sup(\{f\}) - sup(\{fh\})$

8

Knowing frequent positive patterns is not enough

Id	*Transaction*
T_1	*abce*
T_2	*h*
T_3	*abch*
T_4	*abe*
T_5	*acefh*
T_6	*bef*
T_7	*abcef*
T_8	*af*

Let $minSup = 2$. Then:

$\{f\}_{[4]}$, …,

$\{fh\}$ is not frequent

?

$sup(\{f(-h)\}) = sup(\{f\}) - sup(\{fh\}) = ?$

9

Generalized disjunctive rules

- Let Z be an itemset. The expression:

$$X \rightarrow a_1 \vee \ldots \vee a_n$$

is defined a *generalized disjunctive rule based on* Z (and Z is *the base of* $X \rightarrow a_1 \vee \ldots \vee a_n$) if $X \subset Z$ and $\{a_1, \ldots, a_n\} = Z \backslash X$.

- **Example.** Let $Z = \{abc\}$. There are 2^3-1 gen. dis. rules:

$\emptyset \rightarrow a \vee b \vee c$

$a \rightarrow b \vee c$ $\quad$ $b \rightarrow a \vee c$ $\quad$ $c \rightarrow a \vee b$

$ab \rightarrow c$ $\quad$ $ac \rightarrow b$ $\quad$ $bc \rightarrow a$

10

Errors of generalized disjunctive rules

- $sup(X \rightarrow a_1 \vee \ldots \vee a_n)$ is defined as the number of transactions in which X occurs together with a_1 or a_2, or … or a_n.
- $err(X \rightarrow a_1 \vee \ldots \vee a_n)$ is defined as the number of transactions that contain X and do not contain any item in $\{a_1, \ldots, a_n\}$; that is:

X	*a*
X	*b*
X	*c*
X	*d*

$sup(X)$

$sup(X \rightarrow a \vee b \vee c)$

$err(X \rightarrow a \vee b \vee c)$

- $X \rightarrow a_1 \vee \ldots \vee a_n$ is defined as a *certain rule* if $err(X \rightarrow a_1 \vee \ldots \vee a_n) = 0$.

11

Reasoning with certain rules...

- **Example.**

$err(X \rightarrow a \vee b) = 0$

$\Updownarrow$

$sup(X) = sup(X \rightarrow a \vee b)$

$\Updownarrow$

$sup(X) = sup(Xa) + sup(Xb) - sup(Xab)$

$\Updownarrow$

$sup(Xab) = sup(Xa) + sup(Xb) - sup(X)$

	a	
X	*a*	
X	*a*	
X	*a*	*b*
X		*b*
		b

Generalized conclusion.

$err(X \rightarrow \vee Y) = 0$ iff $sup(XY) = (-1)^{|Y|} \times [\Sigma_{Z \subset Y} (-1)^{|Z|-1} \times sup(XZ)]$.

- Thus, if $X \rightarrow \vee Y$ is certain, then $sup(XY)$ is determinable from the supports of proper subsets of XY.

12

Reasoning with certain rules

- **Example.**

$$err(X \rightarrow a \vee b) = 0$$

$$\Downarrow$$

$$err(YX \rightarrow a \vee b) = 0$$

		a	
	X	*a*	
Y	*X*	*a*	
Y	*X*	*a*	*b*
Y	*X*		*b*
			b

- **Generalized conclusion.** If $X \rightarrow \vee Y$ is certain, then $ZX \rightarrow \vee Y$ is also certain and determines a method of calculating $sup(ZXY)$ from the supports of proper subsets of ZXY.
- **Corollary.** If there is a certain rule based on itemset Z, then the supports of all supersets of Z are derivable from their proper subsets.

13

Generalized disjunctive (derivable) sets and generalized disjunction-free (non-derivable) sets

- Itemset X is defined as a *generalized disjunctive set* (or *derivable*) if there is a certain generalized disjunctive rule based on X.
- **Note:** There is $2^{|X|}$-1 generalized disjunctive rules based on X.
- Itemset X is defined as a *generalized disjunction-free set* (or non-derivable) if there is no certain generalized disjunctive rule based on X.

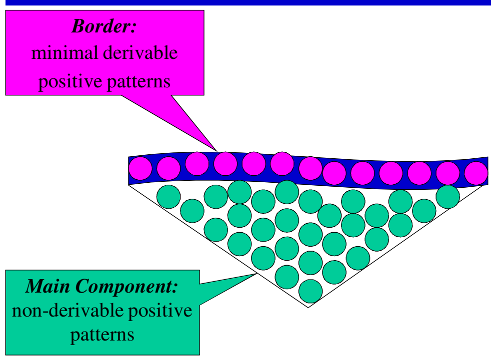
14

Generalized disjunctive (derivable) sets and generalized disjunction-free (non-derivable) sets

- Supersets of a generalized disjunctive (derivable) set are generalized disjunctive (derivable).
- Subsets of a generalized disjunction-free set (non-derivable) are generalized disjunction-free set (non-derivable)

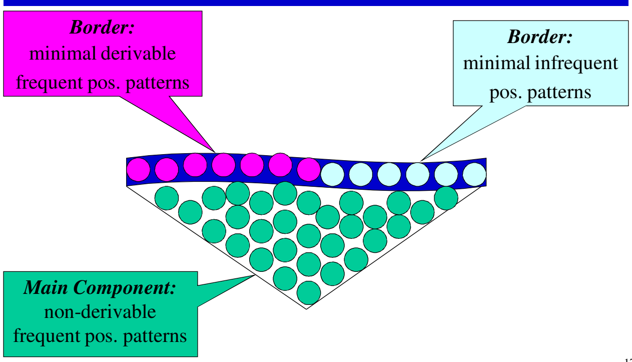
15

Representing all positive patterns



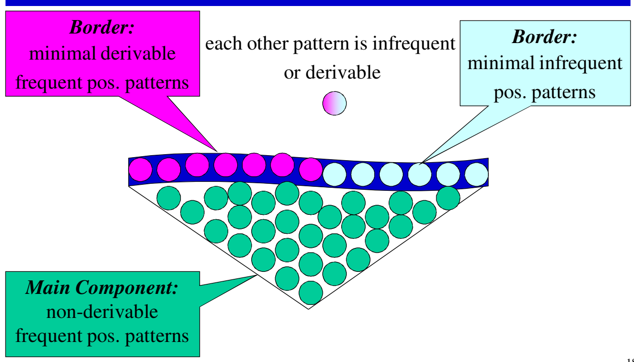
16

GDFSR: Representing all **frequent** positive patterns



17

GDFSR: Representing all frequent positive patterns



18

GDFSR: Deriving frequent and infrequent positive patterns

Example. *Pattern **{abcf}** problem:* We note that $\{abcf\}$ has a subset, e.g. $\{abf\}$, in the infrequent border. This means that all supersets of $\{abf\}$, in particular $\{abcf\}$, are infrequent.

Generalized disjunctive border:

$\{ac\}_{[4,\ c \Rightarrow a]}$ $\{be\}_{[4,\ e \Rightarrow b]}$

Infrequent border:

$\{abf\}$ $\{aef\}$ $\{bcf\}$ $\{cef\}$
$\{bh\}$ $\{eh\}$ $\{fh\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

19

GDFSR: Deriving frequent and infrequent positive patterns

Example. *Pattern **{abce}** problem*. It does not have any subset in the infrequent border, but has a subset, e.g. $\{ac\}$, in the generalized disjunctive border.

Generalized disjunctive border:

$\{ac\}_{[4,\ c \Rightarrow a]}$ $\{be\}_{[4,\ e \Rightarrow b]}$

Infrequent border:

$\{abf\}$ $\{aef\}$ $\{bcf\}$ $\{cef\}$
$\{bh\}$ $\{eh\}$ $\{fh\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

20

GDFSR: Deriving frequent and infrequent positive patterns

Example. *Pattern **{abce}** problem*: Property $c \Rightarrow a$, associated with set $\{ac\}$ implies property $bce \Rightarrow a$ related to set $\{abce\}$.

Generalized disjunctive border:

$\{ac\}_{[4,\ c \Rightarrow a]}$ $\{be\}_{[4,\ e \Rightarrow b]}$

Infrequent border:

$\{abf\}$ $\{aef\}$ $\{bcf\}$ $\{cef\}$
$\{bh\}$ $\{eh\}$ $\{fh\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

21

GDFSR: Deriving frequent and infrequent positive patterns

Example. *Pattern **{abce}** problem*: Property $c \Rightarrow a$, associated with set $\{ac\}$ implies property $bce \Rightarrow a$ related to set $\{abce\}$. Hence, $\boldsymbol{sup}(\{\mathbf{abce}\}) = \boldsymbol{sup}(\{\mathbf{bce}\}) = ?$

Generalized disjunctive border:

$\{ac\}_{[4,\ c \Rightarrow a]}$ $\{be\}_{[4,\ e \Rightarrow b]}$

Infrequent border:

$\{abf\}$ $\{aef\}$ $\{bcf\}$ $\{cef\}$
$\{bh\}$ $\{eh\}$ $\{fh\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

22

GDFSR: Deriving frequent and infrequent positive patterns

Example. *Pattern {**bce**} subproblem*: $\{bce\}$ has subset $\{be\}$ in the generalized disjunctive border.

Generalized disjunctive border:

$\{ac\}_{[4,\ c \Rightarrow a]}$ $\{be\}_{[4,\ e \Rightarrow b]}$

Infrequent border:

$\{abf\}$ $\{aef\}$ $\{bcf\}$ $\{cef\}$
$\{bh\}$ $\{eh\}$ $\{fh\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

23

GDFSR: Deriving frequent and infrequent positive patterns

Example. *Pattern {**bce**} subproblem*: Property $e \Rightarrow b$ associated with $\{be\}$ implies property $ce \Rightarrow b$ related to $\{bce\}$. Hence, $sup(\{bce\}) = \boldsymbol{sup}(\{\mathbf{ce}\}) = ?$

Generalized disjunctive border:

$\{ac\}_{[4,\ c \Rightarrow a]}$ $\{be\}_{[4,\ e \Rightarrow b]}$

Infrequent border:

$\{abf\}$ $\{aef\}$ $\{bcf\}$ $\{cef\}$
$\{bh\}$ $\{eh\}$ $\{fh\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

24

GDFSR: Deriving frequent and infrequent positive patterns

Example. *Pattern {ce} subproblem*: Pattern $\{ce\}$ belongs to the main component, so its support is known: $sup(\{ce\})$ **=** **2**.

Generalized disjunctive border:

$\{ac\}_{[4,\, c \Rightarrow a]}$ $\{be\}_{[4,\, e \Rightarrow b]}$

Infrequent border:

$\{abf\}$ $\{aef\}$ $\{bcf\}$ $\{cef\}$
$\{bh\}$ $\{eh\}$ $\{fh\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

25

GDFSR: Deriving frequent and infrequent positive patterns

Example. *Summarizing pattern {abce} problem*:

***sup*({abce})** = $sup(\{bce\}) = sup(\{ce\})$ = **2**.

Generalized disjunctive border:

$\{ac\}_{[4,\, c \Rightarrow a]}$ $\{be\}_{[4,\, e \Rightarrow b]}$

Infrequent border:

$\{abf\}$ $\{aef\}$ $\{bcf\}$ $\{cef\}$
$\{bh\}$ $\{eh\}$ $\{fh\}$

Main component:

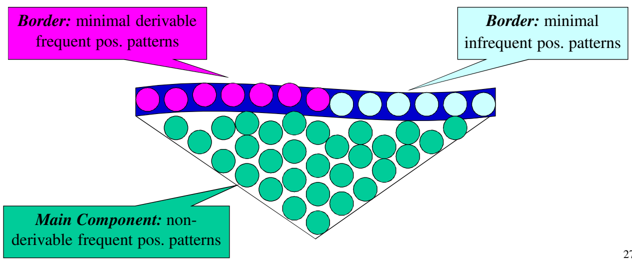
$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

26

GDFSR: Represents all frequent patterns with negation?

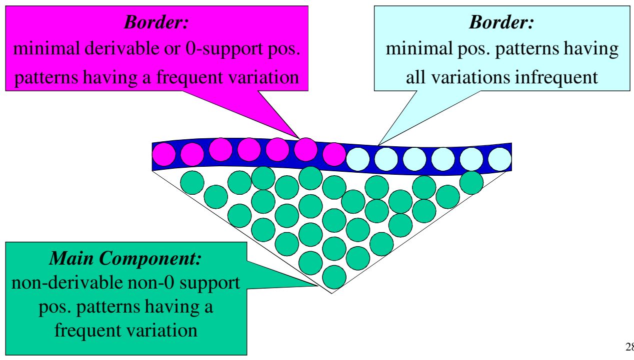
Problem

An infrequent non-derivable positive pattern ○ may have a frequent variation ● with negation



27

GDFLR: Representing all frequent patterns (also with negation)



28

GDFLR: Deriving frequent and infrequent patterns with negation

Example. *Pattern* ***{a(–c)(–e)f}*** *problem:* **{*acef*}** - positive variation of the evaluated pattern - has subset **{*cef*}** in the infrequent border. This means that all supersets of {*cef*} and all their variations, including {*acef*} and {*a*(–*c*)(–*e*)*f*}, are infrequent.

Generalized disjunctive border:

$\{bcf\}_{[1,\, c \Rightarrow b \vee f]}$ $\{bch\}_{[1,\, bh \Rightarrow c]}$ $\{bfh\}_{[0]}$ $\{cfh\}_{[1,\, fh \Rightarrow c]}$
$\{abf\}_{[1,\, f \Rightarrow a \vee b]}$ $\{abh\}_{[1,\, bh \Rightarrow a]}$ $\{aef\}_{[1,\, f \Rightarrow a \vee e]}$ $\{afh\}_{[1,\, fh \Rightarrow a]}$
$\{ac\}_{[4,\, c \Rightarrow a]}$ $\{be\}_{[4,\, e \Rightarrow b]}$ $\{eh\}_{[0]}$

Infrequent border:

$\{cef\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{bh\}_{[1]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$ $\{fh\}_{[1]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

29

GDFLR: Deriving frequent and infrequent patterns with negation

Example. *Pattern* **{*bef*(–*h*)}** *problem.* The positive variation **{*befh*}** of {*bef*(–*h*)} does not have any subset in the infrequent border, so {*bef*(–*h*)} has a chance to be frequent.

Generalized disjunctive border:

$\{bcf\}_{[1,\, c \Rightarrow b \vee f]}$ $\{bch\}_{[1,\, bh \Rightarrow c]}$ $\{bfh\}_{[0]}$ $\{cfh\}_{[1,\, fh \Rightarrow c]}$
$\{abf\}_{[1,\, f \Rightarrow a \vee b]}$ $\{abh\}_{[1,\, bh \Rightarrow a]}$ $\{aef\}_{[1,\, f \Rightarrow a \vee e]}$ $\{afh\}_{[1,\, fh \Rightarrow a]}$
$\{ac\}_{[4,\, c \Rightarrow a]}$ $\{be\}_{[4,\, e \Rightarrow b]}$ $\{eh\}_{[0]}$

Infrequent border:

$\{cef\}$

Main component:

$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{bh\}_{[1]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$ $\{fh\}_{[1]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

30

GDFLR: Deriving frequent and infrequent patterns with negation

Example. *Pattern* **{*bef*(–*h*)}** *problem*:

$sup(\{bef(-h)\}) = \mathbf{sup(\{bef\})} - \mathbf{sup(\{befh\})} = \ldots = \mathbf{2} - \mathbf{0} = \mathbf{2}.$

Generalized disjunctive border:

$\{bcf\}_{[1,\, c \Rightarrow b \vee f]}$ $\{bch\}_{[1,\, bh \Rightarrow c]}$ $\{bfh\}_{[0]}$ $\{cfh\}_{[1,\, fh \Rightarrow c]}$
$\{abf\}_{[1,\, f \Rightarrow a \vee b]}$ $\{abh\}_{[1,\, bh \Rightarrow a]}$ $\{aef\}_{[1,\, f \Rightarrow a \vee e]}$ $\{afh\}_{[1,\, fh \Rightarrow a]}$
$\{ac\}_{[4,\, c \Rightarrow a]}$ $\{be\}_{[4,\, e \Rightarrow b]}$ $\{eh\}_{[0]}$

Infrequent border:

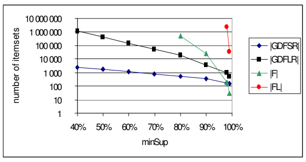
$\{cef\}$

Main component:

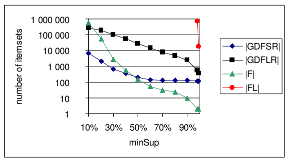
$\{ab\}_{[4]}$ $\{ae\}_{[3]}$ $\{af\}_{[3]}$ $\{ah\}_{[2]}$ $\{bc\}_{[3]}$ $\{bf\}_{[2]}$ $\{bh\}_{[1]}$ $\{ce\}_{[2]}$ $\{cf\}_{[2]}$ $\{ch\}_{[2]}$ $\{ef\}_{[2]}$ $\{fh\}_{[1]}$
$\{a\}_{[6]}$ $\{b\}_{[5]}$ $\{c\}_{[4]}$ $\{e\}_{[4]}$ $\{f\}_{[4]}$ $\{h\}_{[3]}$
$\emptyset_{[8]}$

31

Conciseness of the representation



connect-4 data set



mushroom data set

32

Basic steps in *GDFLR-Apriori* algorithm

Assertion: candidate elements are positive patterns

- Calculate supports of candidates w.r.t. database.
- **Calculate errors of all generalized disjunctive rules for each candidate.**
- **Determine supports of all variations for each candidate.**
- Split candidates into *Main Component* and *Borders* based on errors of rules and supports of variations of candidates.
- Create longer candidates by merging pairs of elements of *Main Component*.
- Prune all new candidates that do not have all proper subsets in *Main Compoment*.

33

Errors of rules and supports of variations

- **Example.**

$err(X \rightarrow a \vee b) = sup(X(-a)(-b))$

X	*a*
X	*b*
X	
X	*d*

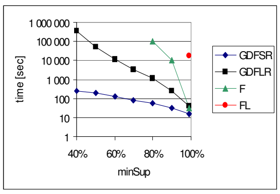
Generalized conclusion.

$err(X \rightarrow a_1 \vee \ldots \vee a_n) = sup(X\,(-a_1) \ldots (-a_n)).$

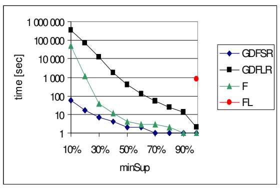
- For a pattern *P*, the set of the errors of all generalized disjunctive rules based on *P* equals the set of the supports of all variations of *P* that are different from *P*.

34

Extraction time



connect-4 data set



mushroom data set

35

Upper bound on length of elements of GDFLR...

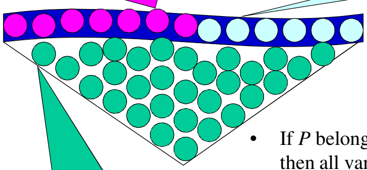
- **Earlier conclusion:** For a pattern *P*, ***the set of the errors of all generalized disjunctive rules*** based on *P* equals ***the set of the supports of all variations*** of *P* that are different from *P*.
- **Corollary 1:** For a **positive** pattern *P*, **the set of the errors of all generalized disjunctive rules** based on *P* equals **the set of the supports of all variations with negation** of *P*.
- **Corollary 2:** The following statements are equivalent for a generalized disjunction-free (non-derivable) positive pattern *P*:
  - ➢all rules based on *P* are not certain
  - ➢the errors of all rules based on *P* are different from 0
  - ➢all variations with negation of *P* have non-zero supports.

36

Upper bound on length of elements of GDFLR...

Border:
minimal derivable or 0-support pos. patterns having a frequent variation

Border:
minimal pos. patterns having all variations infrequent



Main Component:
non-derivable non-0 support pos. patterns having a frequent variation

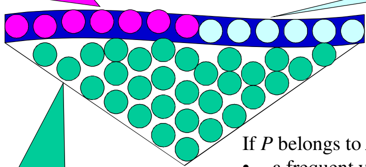
- If P belongs to ***Main Component***, then all variations of P have non-zero supports.
- Hence, each variation of P covers at least one transaction.

37

Upper bound on length of elements of GDFLR...

Border:
minimal derivable or 0-support pos. patterns having a frequent variation

Border:
minimal pos. patterns having all variations infrequent



Main Component:
non-derivable non-0 support pos. patterns having a frequent variation

If P belongs to ***Main Component***, then

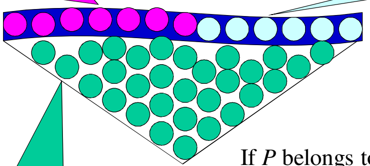
- a frequent variations of P, covers more than *minSup* transactions.
- the remaining $2^{|P|} - 1$ variations of P cover the remaining transactions in D; that is, no more than $|D| - minSup - 1$ transactions.

38

Upper bound on length of elements of GDFLR...

Border:
minimal derivable or 0-support pos. patterns having a frequent variation

Border:
minimal pos. patterns having all variations infrequent



Main Component:
non-derivable non-0 support pos. patterns having a frequent variation

If P belongs to ***Main Component***, then

$$2^{|P|} - 1 \leq |D| - minSup - 1.$$

Hence,

$$|P| \leq \lfloor \log_2(|D| - minSup) \rfloor.$$

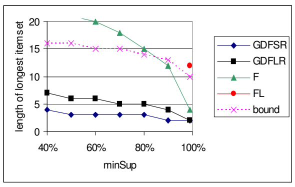
39

Upper bound on length of GDFLR' s elements

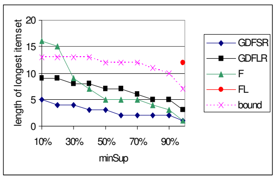
data set	# items	avg. length	std. dev	# records	$minSup / \lvert\mathcal{D}\rvert$	length of longest frequent itemset	$\lfloor \log_2(\lvert\mathcal{D}\rvert - minSup) \rfloor + 1$
chess	76	37	0	3196	20%	23	12
connect-4	129	43	0	67557	10%	29	16
mushroom	119	23	0	8124	0,075%	21	13
pumsb*	7117	50	2	49046	5%	40	16
pumsb	7117	74	0	49046	60%	22	15
gazelle	**498**	**2,5**	**4,9**	**59601**	**0.01%**	**154**	**16**
T10I4D100K	1000	10	3,7	100000	0.025%	11	17
T40I10D100K	1000	40	8,5	100000	0.001%	25	17

40

Upper bound and real length of elemenst



connect-4 data set



mushroom data set

41

Addressed problem:
calculating rule errors /supports of variations

- Calculation of errors of all rules based on X (or supports of all variations of X that are different from X) requires:

$$\Sigma_{n=1..|X|} \binom{|X|}{n} (2^n - 1)$$

accesses to proper subsets of X.

42

Absolute ordering of variations

- n^{th} *variation of pattern* X ($\mathcal{V}_n(X)$) is defined as this variation of X that differs from X on all and only bit positions with value 1 in the binary representation of n ($0 \leq n < 2^{|X|}$).

  For variation $\mathcal{V}_n(X)$, n is called its (*absolute*) *ordering number.*

- **Example.** Let $X = \{abc\}$.

$$\mathcal{V}_5(X) = \mathcal{V}_{2^2+2^0}(X) = \mathcal{V}_{(101)_2}(X) = \{(-a)b(-c)\}$$

5th variation of X

43

Clusters of variations

- k^{th} *cluster* ($\boldsymbol{C}_k(X)$) for pattern X is defined as the set of all variations of X such that k is the leftmost bit position with value 1 in the binary representation of their ordering numbers ($0 \leq k < |X|$).

- **Example.**

basic pattern: X : $\mathcal{V}_{(000)_2}(X)$

0th cluster $C_0(X)$: $\mathcal{V}_{(001)_2}(X)$

1th cluster $C_1(X)$: $\mathcal{V}_{(010)_2}(X)$, $\mathcal{V}_{(011)_2}(X)$

2nd cluster $C_2(X)$: $\mathcal{V}_{(100)_2}(X)$, $\mathcal{V}_{(101)_2}(X)$, $\mathcal{V}_{(110)_2}(X)$, $\mathcal{V}_{(111)_2}(X)$

44

Ordering variations in clusters

- $\mathcal{V}_{2^k+j}(X)$, where $j < 2^k$, is called j^{th} *variation of pattern X in cluster* $C_k(X)$.

cluster $C_2(X) = \{\mathcal{V}_{(100)_2}(X), \mathcal{V}_{(101)_2}(X), \mathcal{V}_{(110)_2}(X), \mathcal{V}_{(111)_2}(X)\}$

absolute ordering:

5th variation of X

relative ordering:

1th variation

in cluster $C_2(X)$

45

Relationships between variations

Let X be a non-empty pattern, $k \in \{0, \ldots, |X|-1\}$ and $j < 2^k$.

- **Note.** j^{th} variation in k^{th} cluster ($\mathcal{V}_{2^k+j}(X)$) differs from j^{th} variation ($\mathcal{V}_j(X)$) only on position k.

$$\Downarrow$$

$$sup(\mathcal{V}_{2^k+j}(X)) = sup(\mathcal{V}_j(X \setminus X[k])) - sup(\mathcal{V}_j(X))$$

- **Conclusion (principle of *GDFLR-SO-Apriori*).** The supports of all variations of X belonging to the same cluster are determinable from:
  - the supports of variations of the same subset of X, and
  - the supports of variations of X with less absolute ordering number.

46

Calculating the supports of variations

X - non-empty pattern, $i \in \{0, \ldots, |X|-1\}$ and $j < 2^i$.

$$\boldsymbol{sup(\mathcal{V}_{2^i+j}(X)) = sup(\mathcal{V}_j(X \setminus \{X[i]\})) - sup(\mathcal{V}_j(X))}$$

$X = \{abc\}$ $\quad sup(\mathcal{V}_{2^0+0}(X)) = sup(\mathcal{V}_0(X \setminus X[0]))$ - $sup(\mathcal{V}_0(X))$

i	j			
0	0	$sup(\{ab(-c)\}) =$	$sup(\{ab\})$	$- sup(\{abc\})$
1	0	$sup(\{a(-b)c\}) =$	$sup(\{ac\})$	$- sup(\{abc\})$
	1	$sup(\{a(-b)(-c)\}) =$	$sup(\{a(-c)\})$	$- sup(\{ab(-c)\})$
2	0	$sup(\{(-a)bc\}) =$	$sup(\{bc\})$	$- sup(\{abc\})$
	1	$sup(\{(-a)b(-c)\}) =$	$sup(\{b(-c)\})$	$- sup(\{ab(-c)\})$
	2	$sup(\{(-a)(-b)c\}) =$	$sup(\{(-b)c\})$	$- sup(\{a(-b)c\})$
	3	$sup(\{(-a)(-b)(-c)\}) =$	$sup(\{(-b)(-c)\})$	$- sup(\{a(-b)(-c)\})$

47

Advanced versus naive calculation of rule errors /supports of variations

- New calculation of supports of all variations of X requires:
  - $|X|$ accesses to ($|X|$-1)-item subsets of X, and
  - knowing the supports of all variations for all these subsets of X.

- Previous calculation of errors of all rules based on X requires:

$$\Sigma_{n=1..|X|} \binom{|X|}{n} (2^n - 1)$$

accesses to proper subsets of X.

48

Experimental results for the connect-4 dataset

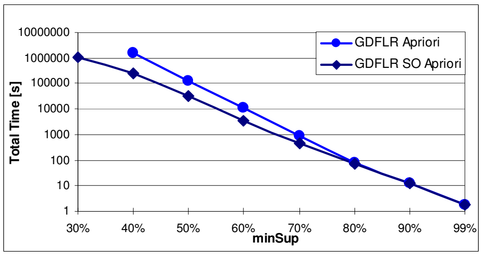


Figure. Duration of *GDFLR-SO-Apriori* and *GDFLR-Apriori* (logarithmic scale)

Experimental results for the mushroom dataset

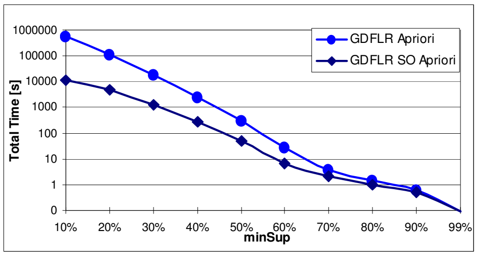


Figure. Duration of *GDFLR-SO-Apriori* and *GDFLR-Apriori* (logarithmic scale)

Experimental results for the connect-4 dataset

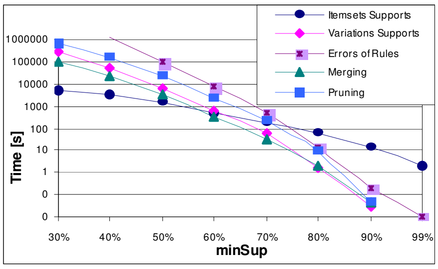


Figure. Duration of particular phases of *GDFLR-SO-Apriori* and *GDFLR-Apriori* (log. scale)

Experimental results for the mushroom dataset

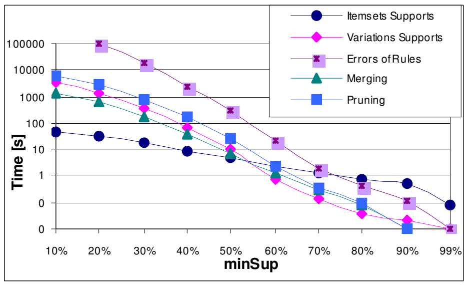


Figure. Duration of particular phases of *GDFLR-SO-Apriori* and *GDFLR-Apriori* (log. scale)

Experimental results for the connect-4 dataset

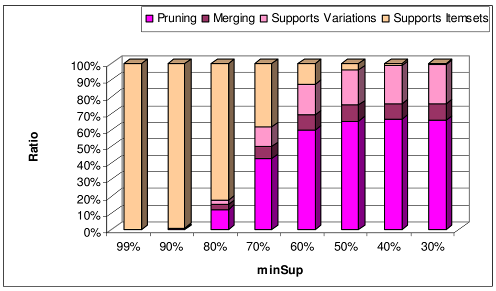


Figure. The proportion of runtime of individual phases of *GDFLR-SO-Apriori*

Experimental results for the mushroom dataset

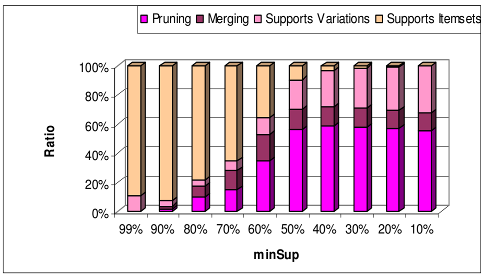


Figure. The proportion of runtime of individual phases of *GDFLR-SO-Apriori*

Experimental results

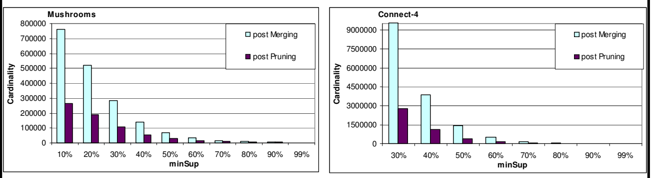
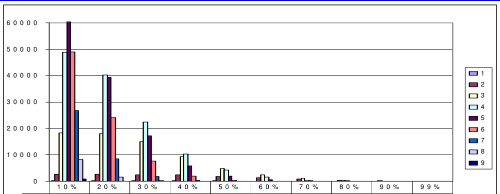


Figure. Number of candidates generated in the merging phase versus number of candidates remained after pruning phase

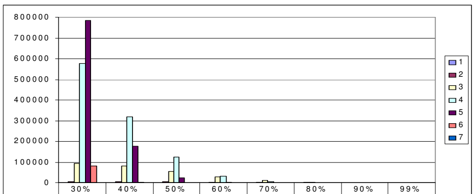
Experimental results for the mushroom dataset



	10%	20%	30%	40%	50%	60%	70%	80%	90%	99%
1	118	118	118	118	118	105	95	80	66	23
2	2579	2579	2566	2360	1837	1244	818	506	203	11
3	18315	18142	14891	9423	4936	2539	1129	477	103	0
4	48774	40205	22367	10198	4205	1630	519	166	44	0
5	62023	39476	17193	5909	2033	576	139	32	0	0
6	49072	24210	7632	2017	549	96	14	0	0	0
7	26874	8592	1833	377	76	4	0	0	0	0
8	8179	1526	194	14	0	0	0	0	0	0
9	946	67	0	0	0	0	0	0	0	0

Figure. Number of patterns of the Main component found in each iteration

Experimental results for the connect-4 dataset



	30%	40%	50%	60%	70%	80%	90%	99%
1	129	129	129	124	114	98	77	28
2	5501	5412	5049	4256	3012	1717	444	12
3	92413	80066	56166	30389	11852	3547	124	0
4	575818	320143	123621	32526	6916	576	0	0
5	786002	177156	23758	1695	73	0	0	0
6	81826	3417	50	0	0	0	0	0
7	57	0	0	0	0	0	0	0

Figure. Number of patterns of the Main component found in each iteration

Summary

- The set of all positive patterns can be treated as a lossless representation of all frequent patterns, nevertheless it is not concise.
- The set of all frequent positive patterns neither guarantees derivation of all frequent patterns with negation, nor is concise in practice.
- GDFLR representations, which consists of a subset of positive patterns, admits derivation of both all frequent positive patterns and patterns with negation.

Summary

- The experiments carried out on real benchmark data sets show that GDFLR is by several orders of magnitude more concise than all frequent patterns.
- The length of longest elements in GDFLR depends logarithmically on the number of records in the data set, which implies that the number of scans of the data set carried out by *Apriori*-like algorithms discovering GDFLR also depends logarithmically on the number of records in the data set.

Summary

- The *GDFLR-SO-Apriori* algorithm is faster than the *GDFLR-Apriori* algorithm by up to two orders of magnitude for low support threshold values.
- The speed-up is obtained as a result of proposing an ordering of variations and calculating their supports in accordance with that order, which admits re-use of the partial results.

References…

- Marzena Kryszkiewicz: Concise Representation of Frequent Patterns Based on Disjunction-Free Generators. ICDM 2001: 305-312
- Marzena Kryszkiewicz, Marcin Gajek: Concise Representation of Frequent Patterns Based on Generalized Disjunction-Free Generators. PAKDD 2002: 159-171
- Marzena Kryszkiewicz: Concise Representations of Frequent Patterns and Association Rules, Warsaw: Publishing House of Warsaw University of Technology (2002)

61

References…

- Marzena Kryszkiewicz, Katarzyna Cichon: Support Oriented Discovery of Generalized Disjunction-Free Representation of Frequent Patterns with Negation. PAKDD 2005: 672-682
- Marzena Kryszkiewicz: Non-Derivable Item Set and Non-Derivable Literal Set Representations of Patterns Admitting Negation. DaWaK 2009: 138-150
- Marzena Kryszkiewicz: Reasoning about Frequent Patterns with Negation. Encyclopedia of Data Warehousing and Mining 2009: 1667-1674

62

Thank you for your attention

63