Local Constraint-Based Mining and Set Constraint Programming for Pattern Discovery

Mehdi Khiari, Patrice Boizumault, Bruno Crémilleux

Greyc, Université de Caen Basse-Normandie

GREYC

September 7, 2009

ECML/PKDD, Workshop LeGo 2009

Pattern Flooding: a well-known limitation of local patterns

examples of local patterns:

- regularities: frequent patterns, area (can be used to discover synexpression groups),...
- contrasts: emerging patterns,. . .

$\Rightarrow$ in practice, usual techniques provide an overwhelming number of patterns

How to reduce/summarize local patterns?

1. exact/approximate condensed representations of patterns (Pasquier et al. ICDT'99, Boulicaut et al. DMKD'03, Calders et al. LNAI'05, Casali et al. DaWaK'05, Soulet et al. DMKD'08,. . .)
2. the constraint-based paradigm:
   - a lot of contributions on local patterns (Ng et al. SIGMOD'98, Bucilla et al. SIGKDD'02, De Raedt et al. ICDM'02, Besson et al. IDA'05, Soulet et al. PAKDD'05, . . .)
   - integrating external resources and background knowledge (Klema et al. ISB'08)

How to reduce/summarize local patterns?

3. selecting patterns on the basis of their usefulness in the context of the other selected patterns:
    - pattern teams (Knobbe et al., PKDD'06)
    - constraint-based pattern set mining (De Raedt et al., SDM'07), the chosen few (Bringmann et al., ICDM'07)
4. compression of the dataset by exploiting the MDL Principle (Siebes et al., SDM'06)
5. using constraint programming (0/1 Linear Programming) (De Raedt et al., KDD'08, Nijssen et al., KDD'09)

Global Constraints

Definition (Global constraint)

A constraint q is said *global* if several patterns have to be compared to check if q is satisfied or not.

In this talk, a global constraint is a n-ary constraint

Example of global constraint

Trans.	Items					
o_1	A	B			c_1	
o_2	A	B			c_1	
o_3			C	D	c_1	
o_4	A	B		D	c_1	
o_5	A	B		D	c_1	
o_6	A	B		D	c_1	
o_7			C			c_2
o_8	A	B	C	D		c_2
o_9				D		c_2

the exception rules constraint (Suzuki 2002)

$$exception(X \rightarrow \neg I) \equiv \begin{cases} true & \text{if } \exists Y \in \mathcal{L}_{\mathcal{I}} \text{ such that } Y \subset X, \text{ one have} \\ & \qquad (X \backslash Y \rightarrow I)^a \wedge (X \rightarrow \neg I)^b \\ false & \text{otherwise} \end{cases}$$

[a] common sense rule: frequent + high confidence value

[b] exception rule: rare + very high confidence value

Example of global constraint

Trans.	Items					
o_1	A	B			c_1	
o_2	A	B			c_1	
o_3			C	D	c_1	
o_4	A	B		D	c_1	
o_5	A	B		D	c_1	
o_6	A	B		D	c_1	
o_7			C			c_2
o_8	A	B	C	D		c_2
o_9				D		c_2

$$\begin{cases} AB \longrightarrow c_1 \\ ABC \longrightarrow \neg c_1 \end{cases}$$

the exception rules constraint (Suzuki 2002)

$$exception(X \to \neg I) \equiv \begin{cases} true & \text{if } \exists Y \in \mathcal{L}_{\mathcal{I}} \text{ such that } Y \subset X, \text{ one have } (X \backslash Y \to I)^a \wedge (X \to \neg I)^b \\ false & \text{otherwise} \end{cases}$$

[a] common sense rule: frequent + high confidence value

[b] exception rule: rare + very high confidence value

Motivations

- Constraint programming:
  - A powerful declarative paradigm for solving difficult combinatorial problems,
  - Efficient filtering and solving techniques.
- Set CSPs
  - Variables $\leftrightarrow$ Unknown patterns,
  - Domains $\leftrightarrow 2^{\mathcal{I}}$ (where $\mathcal{I}$ is the set of all the items in the data set)
  - Handling of set constraints ($\subset$, $\cup$) (local and global).

$\Rightarrow$ Investigating the links between data mining and Set Constraint Satisfaction Problems (Set CSPs) is a promising approach.

Outline

Set Intervals

Definition (Set Interval)

Let lb and ub be two sets such that $lb \subset ub$, the set interval $[lb..ub]$ is defined as follows:
$[lb..ub] = \{E \text{ such that } lb \subseteq E \text{ and } E \subseteq ub\}$.

Examples

- $[\{1\}..\{1,2,3\}]=\{\{1\},\{1,2\},\{1,3\},\{1,2,3\}\}$
- $[\{\}..\{1,2,3\}] = 2^{\{1,2,3\}}$

Set CSPs

Definition (Set CSP)

A set constraint satisfaction problem (set CSP) is a 3-uple $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ where $\mathcal{C} = \{c_1, ..., c_m\}$ is a set of constraints associated to a set $\mathcal{X} = \{X_1, ..., X_n\}$ of variables. For each variable X_i, an initial domain of set intervals (or union of set intervals) D_{X_i} is given and $D = \{D_{X_i}, ..., D_{X_n}\}$.

Example of a set CSP

Example

We have to assign sets of radio frequencies to two transmitters according to some constraints. Available frequencies are $\{1, 2, 3, 4\}$ for the first transmitter and $\{3, 4, 5, 6\}$ for the second one.

$\Rightarrow$ set CSP $(\mathcal{X}, \mathcal{D}, \mathcal{C})$, where:

- $\mathcal{X} = \{t_1, t_2\}$ where t_1 and t_2 are the two transmitters.
- $D(t_1) = [\{\} \; .. \; \{1, 2, 3, 4\}]$ $D(t_2) = [\{\} \; .. \; \{3, 4, 5, 6\}]$

Example of a set CSP

- two radio frequencies have to be assigned to each transmitter:
  - $c_1 \quad \mid t_1 \mid = 2$
  - $c_2 \quad \mid t_2 \mid = 2$
- both transmitters do not share frequencies:
  - $c_3 \quad t_1 \cap t_2 = \emptyset$
- two frequencies within a transmitter must have at least a distance equals to 2:
  - $c_4 \quad \forall v_1, v_2 \in t_i, \;\; abs(v_1 - v_2) \geq 2 \quad i = 1, 2$
- the first transmitter requires the frequency 3:
  - $c_5 \quad 3 \in t_1$
- the second transmitter requires the frequency 4:
  - $c_6 \quad 4 \in t_2$

GREYC

Example of a set CSP

Set CSP $(\mathcal{X}, \mathcal{D}, \mathcal{C})$, where:

- $\mathcal{X} = \{t_1, t_2\}$ where t_1 and t_2 are the two transmitters.
- $D(t_1) = [\{\} \ .. \ \{1,2,3,4\}]$, $D(t_2) = [\{\} \ .. \ \{3,4,5,6\}]$

- $\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5, c_6\}$
  - c_1 $\mid t_1 \mid = 2$
  - c_2 $\mid t_2 \mid = 2$
  - c_3 $t_1 \cap t_2 = \emptyset$
  - c_4 $\forall v_1, v_2 \in t_i$, $\mid v_1 - v_2 \mid \geq 2 \quad i = 1,2$
  - c_5 $3 \in t_1$
  - c_6 $4 \in t_2$

$\Rightarrow$ A unique solution: $t_1 = \{1,3\}$ and $t_2 = \{4,6\}$.

Filtering rules for Set CSPs

Let $D_x = [a_x..b_x]$ and $D_y = [a_y \,..\, b_y]$ two domains represented by set intervals and D'_x and D'_y the filtered domains.

Constraint: $X \subset Y$

Filtering rule: **if** $a_x \subset b_y$ **then**

$$D'_x = [a_x \,..\, b_x \cap b_y]$$
$$D'_y = [a_x \cup a_y \,..\, b_y]$$

else

$$D'_x = \emptyset, D'_y = \emptyset$$

$X \subset Y$

$$D_x = [\{1,2\}..\{1,2,3,4\}], D_y = [\{1\}..\{1,2,3\}]$$
$$D'_x = [\{1,2\}..\{1,2,3\}], D'_y = [\{1,2\}..\{1,2,3\}]$$

Filtering rules for Set CSPs

Let $D_x = [a_x..b_x]$, $D_y = [a_y \,..\, b_y]$ and $D_z = [a_z \,..\, b_z]$ three domains represented by set intervals and D'_x, D'_y and D'_z the filtered domains.

Constraint: $Z = X \cap Y$

Filtering rule: **if** $(b_x \cap b_y) \subset b_z$ and $(b_x \cap b_y) \neq \emptyset$ **then**

$$D'_x = [a_x \cup a_z \,..\, b_x \setminus ((b_x \cap a_y) \setminus b_z]$$
$$D'_y = [a_y \cup a_z \,..\, b_y \setminus ((b_y \cap a_x) \setminus b_z]$$
$$D'_z = [a_z \cup (a_x \cap a_y) \,..\, b_z \cap b_x \cap b_y]$$

else

$$D'_x = D'_y = D'_z = \emptyset$$

GREYC

Outline

GREYC

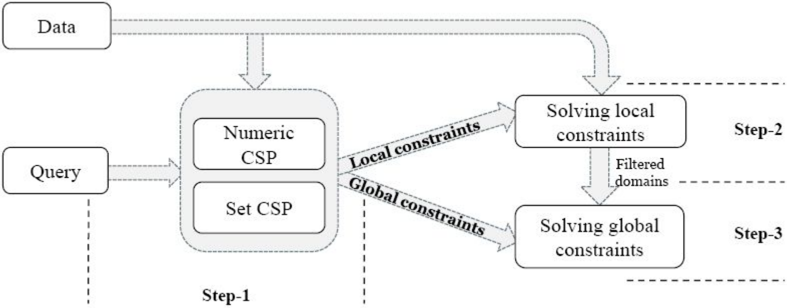
Set CSPs for Pattern Discovery: our aproach

Our approach is based on three major points:

1. the wide possibilities of modelization and resolution given by the CSPs
    - set CSPs
    - numeric CSPs
2. the recent progress on mining local patterns
3. local constraints can be solved before and regardless global constraints.

GREYC

General overview of our 3-steps method



Step-1: Modeling the query as CSPs

1. Set CSP $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ where:
   - $\mathcal{X} = \{X_1, ..., X_n\}$. Each variable X_i represents an unknown itemset.
   - $\mathcal{D} = \{D_{X_1}, ..., D_{X_n}\}$. The initial domain of each variable X_i is the set interval $[\{\} \;..\; \mathcal{I}]$.
   - $\mathcal{C}$ is a conjunction of set constraints by using set operators $(\cup, \cap, \setminus, \in, \notin, ...)$
2. Numeric CSP $\mathcal{P}' = (\mathcal{F}, \mathcal{D}', \mathcal{C}')$ where:
   - $\mathcal{F} = \{F_1, ..., F_n\}$. Each variable F_i is the frequency of the itemset X_i.
   - $\mathcal{D}' = \{D_{F_1}, ..., D_{F_n}\}$. The initial domain of each variable F_i is the integer interval $[1 \;..\; nb]$.
   - $\mathcal{C}'$ is a conjunction of arithmetic constraints.

GREYC

Example of modeling

the exception rules constraint

$$exception(X \rightarrow \neg I) \equiv \begin{cases} true & \text{if } \exists Y \in \mathcal{L}_{\mathcal{I}} \text{ such that } Y \subset X \text{, one have} \\ & \qquad (X \backslash Y \rightarrow I) \wedge (X \rightarrow \neg I) \\ false & \text{otherwise} \end{cases}$$

the exception rules constraint (2)

$$exception(X \rightarrow \neg I) \equiv \begin{cases} \exists Y \subset X \text{ such that :} \\ \quad freq((X \setminus Y) \sqcup I) \geq \gamma_1 \\ \wedge\ (freq(X \setminus Y) - freq((X \setminus Y) \sqcup I)) \leq \delta_1 \\ \wedge\ freq(X \sqcup \neg I) \leq \gamma_2 \\ \wedge\ (freq(X) - freq(X \sqcup \neg I)) \leq \delta_2 \end{cases}$$

γ_1 and γ_2: frequency thresholds

δ_1 and δ_2: confidence thresholds

the exception rules constraint (2)

$$exception(X \rightarrow \neg I) \equiv \begin{cases} \exists Y \subset X \text{ such that :} \\ \quad freq((X \setminus Y) \sqcup I) \geq \gamma_1 \\ \wedge\ (freq(X \setminus Y) - freq((X \setminus Y) \sqcup I)) \leq \delta_1 \\ \wedge\ freq(X \sqcup \neg I) \leq \gamma_2 \\ \wedge\ (freq(X) - freq(X \sqcup \neg I)) \leq \delta_2 \end{cases}$$

The CSP variables are defined as follows:

- Set variables $\{X_1, X_2, X_3, X_4\}$ representing unknown itemsets:
  - $X_1 : X \setminus Y$,
  - $X_2 : (X \setminus Y) \sqcup I$ (common sense rule),
  - $X_3 : X$,
  - $X_4 : X \sqcup \neg I$ (exception rule).
- Integer variables $\{F_1, F_2, F_3, F_4\}$ representing their frequency values (variable F_i denotes the frequency of the itemset X_i).

GREYC

Constraints	CSP formulation	Local	Global
	$F_2 \geq \gamma_1$	$\times$	
	$\wedge$		
$freq((X \setminus Y) \sqcup I) \geq \gamma_1$	$I \in X_2$	$\times$	
	$\wedge$		
	$X_1 \subsetneq X_3$		$\times$
	$F_1 - F_2 \leq \delta_1$		$\times$
$freq(X \setminus Y) - freq((X \setminus Y) \sqcup I) \leq \delta_1$	$\wedge$		
	$X_2 = X_1 \sqcup I$		$\times$
	$F_4 \leq \gamma_2$	$\times$	
$freq(X \sqcup \neg I) \leq \gamma_2$	$\wedge$		
	$\neg I \in X_4$	$\times$	
	$F_3 - F_4 \leq \delta_2$		$\times$
$freq(X) - freq(X \sqcup \neg I) \leq \delta_2$	$\wedge$		
	$X_4 = X_3 \sqcup \neg I$		$\times$

Table: Exception rules modeled as CSP constraints

Steps 2 & 3: From Local to Global

Step-2: Solving local constraints

```
---------------
./music-dfs -i donn.bin -q "{c1} subset X2 and freq(X2)>=4;"
X2 in [{A, c1}..{A, c1, B}] U {B, c1} -- F2 = 5 ;
X2 in {D, c1} -- F2 = 4
---------------
```

Step-3: Solving global constraints

```
---------------
[eclipse 1]:
?- exceptions(X1, X2, X3, X4).
Sol1 : X1 = {A,B}, X2={A,B,c1}, X3={A,B,C}, X4={A,B,C,c2};
.../...
---------------
```

Outline

Number of pairs of rules

(*postoperative-patient-data:* 90×23)

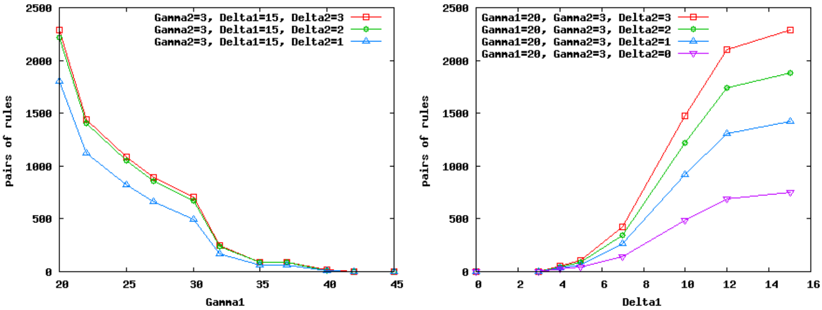
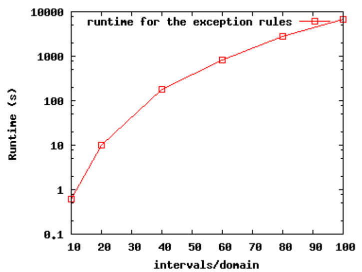


Figure: Number of rules according to γ_1 (left) and δ_1 (right)

- *Correct and complete* set of all pairs of exception rule
- Easy control of the quality (confidence and frequency) of the rules

GREYC

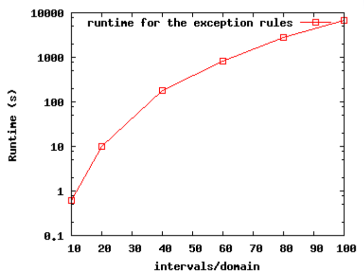
Runtime according to the number of intervals



- **Problem**: unsuitable set intervals union operator: $[lb_1 \,..\, ub_1] \bigcup_{interval} [lb_2 \,..\, ub_2] = [lb_1 \cap lb_2 \,..\, ub_1 \cup ub_2]$.

- $\Rightarrow [\{1\}..\{1,2\}] \bigcup_{interval} [\{3\}..\{3,4\}] = [\{\}..\{1,2,3,4\}] = 2^{\{1,2,3,4\}}$
- Expected result: $\{\{1\},\{1,2\},\{3\},\{3,4\}\}$

Runtime according to the number of intervals



- **Problem**: unsuitable set intervals union operator: $[lb_1 \; .. \; ub_1] \bigcup_{interval} [lb_2 \; .. \; ub_2] = [lb_1 \cap lb_2 \; .. \; ub_1 \cup ub_2]$.

- **Solution:** Search is successively performed upon each Interval
- $\Rightarrow$ Nevertheless, we do not fully profit from filtering.

Outline

GREYC

Conclusion

- A new approach for dicovering patterns under global constraints,
- Takes benefit from the recent progress on mining local patterns,
- Flexible way for modeling several global constraints,
- Complete and sound approach.

Outline

GREYC

Discovering synexpression groups

- $\exists X_1, \ldots X_k$ **(k unfixed)** such that

  where min_{area} denotes the minimal area and α is a threshold given by the user to fix the minimal overlapping between the local patterns.

- we can now solve:
  $\exists X_1, \ldots X_k$ **(k fixed)** such that

Constraints	Local	Global

Discovering synexpression groups

- $\exists X_1, ...X_k$ **(k unfixed)** such that
  $\forall_{1 \leq i \leq k}$, $area(X_i) > min_{area}$

  where min_{area} denotes the minimal area and α is a threshold given by the user to fix the minimal overlapping between the local patterns.

- we can now solve:
  $\exists X_1, ...X_k$ **(k fixed)** such that
  $\forall_{1 \leq i \leq k}$, $area(X_i) > min_{area}$

Constraints	Local	Global
$\forall i \in \{1..n\} area(X_i) > min_{area}$	$\times$	

GREYC

Discovering synexpression groups

- $\exists X_1, ...X_k$ **(k unfixed)** such that
  $\forall_{1 \leq i \leq k}$, $area(X_i) > min_{area}$
  $\wedge(\forall_{1 \leq i < j \leq k}$, $area(X_i \cap X_j) > \alpha \times min_{area})$

  where min_{area} denotes the minimal area and α is a threshold given by the user to fix the minimal overlapping between the local patterns.

- we can now solve:
  $\exists X_1, ...X_k$ **(k fixed)** such that
  $\forall_{1 \leq i \leq k}$, $area(X_i) > min_{area}$
  $\wedge(\forall_{1 \leq i < j \leq k}$, $area(X_i \cap X_j) > \alpha \times min_{area})$

Constraints	Local	Global
$\forall i \in \{1..n\} area(X_i) > min_{area}$	$\times$	
$area(X_i \cap X_j) > \alpha \times min_{area}, (1 \leq i < j \leq k)$		$\times$

Discovering synexpression groups

- $\exists X_1, ... X_k$ **(k unfixed)** such that
  $\forall_{1 \leq i \leq k}$, $area(X_i) > min_{area}$
  $\wedge (\forall_{1 \leq i < j \leq k}, area(X_i \cap X_j) > \alpha \times min_{area})$
  $\wedge \nexists Z, (area(Z) > min_{area} \wedge \forall_{1 \leq i \leq k}, area(X_i \cap Z) > \alpha \times min_{area})$

  where min_{area} denotes the minimal area and α is a threshold given by the user to fix the minimal overlapping between the local patterns.

- we can now solve:
  $\exists X_1, ... X_k$ **(k fixed)** such that
  $\forall_{1 \leq i \leq k}$, $area(X_i) > min_{area}$
  $\wedge (\forall_{1 \leq i < j \leq k}, area(X_i \cap X_j) > \alpha \times min_{area})$

Constraints	Local	Global
$\forall i \in \{1..n\} area(X_i) > min_{area}$	$\times$	
$area(X_i \cap X_j) > \alpha \times min_{area}, (1 \leq i < j \leq k)$		$\times$

Discovering synexpression groups

- $\exists X_1, ... X_k$ **(k unfixed)** such that
  $\forall_{1 \leq i \leq k}$, $area(X_i) > min_{area}$
  $\wedge(\forall_{1 \leq i < j \leq k}$, $area(X_i \cap X_j) > \alpha \times min_{area})$
  $\wedge\ \forall Z$, $(area(Z) \leq min_{area} \vee \exists_{1 \leq i \leq k}$, $area(X_i \cap Z) \leq \alpha \times min_{area})$

  where min_{area} denotes the minimal area and α is a threshold given by the user to fix the minimal overlapping between the local patterns.

- we can now solve:
  $\exists X_1, ... X_k$ **(k fixed)** such that
  $\forall_{1 \leq i \leq k}$, $area(X_i) > min_{area}$
  $\wedge(\forall_{1 \leq i < j \leq k}$, $area(X_i \cap X_j) > \alpha \times min_{area})$

Constraints	Local	Global
$\forall i \in \{1..n\} area(X_i) > min_{area}$	$\times$	
$area(X_i \cap X_j) > \alpha \times min_{area}, (1 \leq i < j \leq k)$		$\times$

GREYC

Future work

- Introducing the universal quantification ($\forall$) that classic CSPs are unable to manage $\Rightarrow$ Quantified CSPs (Bordeaux et al. CP'02),
- Solving CSPs with unknown number of variables,
- Implementing a new set interval union operator in the kernel of the solver,
- Using a non exact condensed representation to reduce the number of produced intervals,